


2780/3780 SIMULATOR SYSTEM REFERENCE MANUAL

BFC 5038

BASIC/FOUR and  are registered trademarks
of BASIC/FOUR CORPORATION
COPYRIGHT 1978 © by BASIC/FOUR CORPORATION All rights reserved.

basic / four corporation
an **MAI** company



All specifications are subject to change without notice.

PROPRIETARY INFORMATION

The information contained herein is proprietary to and considered a trade secret of Basic/Four Corporation and shall not be reproduced in whole or in part without the written authorization of Basic/Four Corporation.

TABLE OF CONTENTS

| | Page |
|---|-------------|
| INTRODUCTION | 1 |
| SECTION 1 2780/3780 SIMULATOR | 3 |
| Program Function | 3 |
| Operating Characteristics | 3 |
| VDT Format | 4 |
| Define Terminal Characteristics | 6 |
| File Queue Definition | 6 |
| User Data File Interface | 11 |
| Software Structure | 15 |
| SECTION 2 QUEUE FILE UTILITY | 17 |
| Program Function | 17 |
| Operating Characteristics | 17 |
| SECTION 3 SIGNON FILE UTILITY | 23 |
| Program Function | 23 |
| Operating Characteristics | 23 |
| System Impact | 25 |
| SECTION 4 JOB CONTROL LANGUAGE UTILITY | 27 |
| Program Function | 27 |
| Operating Characteristics | 27 |
| REFERENCES | 29 |

| | | |
|-----------------------|---|------------------|
| DATE 8/1/78 | TITLE 2780/3780 SIMULATOR SYSTEM REFERENCE MANUAL TABLE OF CONTENTS | PAGE i |
|-----------------------|---|------------------|

LIST OF ILLUSTRATIONS

| Figure | | Page |
|---------------|---|-------------|
| 1-1 | Initial Display Format | 3 |
| 1-2 | Message Format | 4 |
| 1-3 | Status Messages | 5 |
| 1-4 | Terminal Characteristics | 6 |
| 1-5 | File Queue Definition (Receive) | 6 |
| 1-6 | File Queue Definition (Transmit) | 6 |
| 1-7 | Operation of Receive Queue | 7 |
| 1-8 | Operation of Transmit Queue | 7 |
| 1-9 | Signon Definition | 8 |
| 1-10 | Signon Definition (Double Card) | 8 |
| 1-11 | Start Communications | 8 |
| 1-12 | Set Date/Time | 11 |
| 2-1 | Program Mode Selection Menu | 17 |
| 2-2 | Generate Transmit Queue | 17 |
| 2-3 | Generate Receive Queue | 18 |
| 2-4 | Add Entry To Transmit Queue | 19 |
| 2-5 | Entry Number and File Name (Modify) | 19 |
| 2-6 | Selected Record Contents (Modify) | 20 |
| 2-7 | Entry Number and File Name (Delete) | 20 |
| 2-8 | Selected Record Contents (Delete) | 20 |
| 2-9 | Modify Receive Queue | 21 |
| 2-10 | Transmit Queue Display | 22 |
| 2-11 | Receive Queue Display | 22 |
| 3-1 | Function Menu | 23 |
| 3-2 | Add To Existing File | 23 |
| 3-3 | Modify Existing Entry | 23 |
| 3-4 | Delete Existing Entry | 24 |
| 3-5 | Signon Image Display | 24 |
| 3-6 | Signon Image | 24 |

LIST OF TABLES

| Table | | Page |
|--------------|----------------------------------|-------------|
| 1 | ASCII to EBCDIC Conversion | 12 |
| 2 | EBCDIC to ESCII Conversion | 12 |
| 3 | Horizontal Print Control | 13 |
| 4 | Vertical Print Control | 13 |

INTRODUCTION

Distributed Data Processing (DDP) is a multiple systems approach to information handling that offers direct cost savings through improved management control, fast data response, and a high degree of reliability. The Basic Four 2780/3780 Simulator System is a DDP system that provides an organization with the capability of handling its own internal transactions independent of the host processor or other satellite systems, while still being able to share data with those systems.

This document is a reference manual for the 2780/3780 Simulator System. Program function and operating characteristics of the 2780/3780 Simulator are fully detailed in Section 1, as are various modes of operation available to support user needs. Section 2 describes the queue file utility that generates and modifies data files used by the simulator as queues

for the reception and transmission of data. Section 3 describes the SIGNON file utility which is used to specify the content and number of signon images required to establish communications between a remote terminal and a host mainframe. Section 4 defines the Job Control Language (JCL) utility which provides file management of 80-character records used to transmit JCL from a Basic Four computer system to a mainframe computer.

A list of references has been included that imparts more complete information on elementary data communications and the IBM 2780/3780 terminals.

The Appendix describes the various steps and procedures that are undertaken at a terminal to fully setup and execute the simulator program.

| | | |
|-----------------------|--|------------------|
| DATE 8/1/78 | TITLE 2780/3780 SIMULATOR SYSTEM REFERENCE MANUAL INTRODUCTION | PAGE 1 |
|-----------------------|--|------------------|

SECTION 1

2780/3780 SIMULATOR

This section describes the function and operation of the 2780/3780 Simulator as the major software component of an integrated distributed data processing package. Certain details concerning the operation of the simulator as a Remote Job Entry terminal which are functions of a particular host main frame are specific to those mainframes and, therefore, not covered in this manual.

Because of the extensive set of options offered, the user may tailor the 2780/3780 Simulator to satisfy requirements of his particular network.

PROGRAM FUNCTION

The 2780/3780 Simulator package allows an appropriate Basic Four processing system to communicate with a host processor or other remote terminals that conform to the IBM 2780/3780 Data Transmission Terminal EBCDIC protocol.

As a Remote Job Entry terminal, a Basic Four system appears to the host processor as an 80-column card reader. Data transmitted from the Basic Four system are interpreted as Job Control Language (JCL), program files, and data sets. These in turn may be used to control job scheduling and execution, data set manipulation, and host system resource allocation. The host processor outputs files to the Basic Four system as though it were a 132-column printer and an 80-column card punch. Files consist of status and logging information sent at the host system's discretion and results of previously submitted jobs. Routing to printer or punch is controlled through JCL and program statements.

Communications from a Basic Four system to another 2780/3780 remote terminal consists of simple exchanges of files required since each side of the communication link is assumed to be a card reader/punch and line printer.

Various user needs are supported by several modes of operation. The standard mode supports the following basic 2780 features:

- Two records per transmission
- Horizontal tabulation
- Auto turnaround
- Point-to-point network
- Vertical tabulation
- Auto Answer

Other operating modes can provide the following features:

- Single record per transmission

- 2780 multiple records per transmission block (up to 7)
- Transparency
- 3780 protocol
- 3780 transparency with extended buffer

The simulation is controlled by menu-driven inquiry/responses. These sequences provide the operator with commands necessary to select protocol and options, define file queues, and to start or stop the data transfer.

Data transfer is controlled by file queues which are built by a supplied utility. These queues (one for transmit and one for receive) specify by file name and device type where the data are to be transferred to/from. This allows the user to spool received data to multiple physical files and to concatenate multiple data sets for transmission.

OPERATING CHARACTERISTICS

The following paragraphs describe the operation of the program and the interaction between the system and the operator.

The simulator is activated by the operator using the key-in RUN "*CP". The system then loads the simulator from the disc and initiates its execution. The simulator clears the VDT and initializes the display format as shown in **Figure 1-1**.

```

LINE...                XMT QUEUE FILE..
MODE..                 XMT BATCH...
ACTIVE DATA FILE..   PRINT DEVICE..
ACTIVE DEVICE..       ALT PRINT DEVICE..
LAST REC NUMBER..     PUNCH DEVICE..
RCV QUEUE FILE..      ALT PUNCH DEVICE..
**STATUS MESSAGES**  SIGNON REC...
1)
2)
3)
4)
5)

                                MODE MENU
1) DEFINE TERMINAL CHARACTERISTICS
2) FILE QUEUE DEFINITION
3) SIGNON DEFINITION
4) START COMMUNICATIONS
5) SET DATE/TIME
ENTER MODE NUMBER DESIRED ! ■
```

Figure 1-1. Initial Display Format

VDT FORMAT

The VDT format consists of a single, fixed template maintained through all phases of simulator operation and a number of variable fields controlled by the operator-simulator dialogue and real-time communications events. This format is divided into three functional areas.

1. Simulator Real-Time Operational Status

The first area of the VDT display is the real-time operational status of the simulator shown in the fields described below:

LINE—The current status of the communications link (OPEN/CLOSED). The line OPEN condition is maintained for as long as the modems at both ends of the communications link are off-hook and in "data" mode.

MODE—Direction of communications with respect to the simulator (INACTIVE/TRANSMIT/RECEIVE).

ACTIVE DATA FILE—Name of the file which is currently being transmitted from, or received into/or "NONE" during INACTIVE mode. Names come from the transmit or receive file queues and conform to BASIC file or device name requirements.

ACTIVE DEVICE—Device corresponding to ACTIVE DATA FILE (NONE/DISC/PRINTER 1/CONSOLE/PRINTER 2/MAG TAPE).

LAST REC NUMBER—A count of the number of records transmitted or received in a logical file (0000-9999). The count will not reset between chained files.

RCV QUEUE FILE—File name of operator/selected receive queue.

XMT QUEUE FILE—File name of operator selected transmit queue.

XMT BATCH—A count of the files transmitted during a communications session (000-999). If several files are chained, the count will increment only after the last file in the chain is transmitted.

The following four fields are all filled from information contained in the operator-selected receive queue.

Possible values are (NONE/DISC/PRINTER1/CONSOLE/PRINTER 2/MAG TAPE).

PRINT DEVICE—Device to receive files which contain 2780/3780 printer routing or those which contain no routing information.

ALT PRINT DEVICE—Alternate device which is automatically selected by the simulator if an error or end of file condition occurs on device receiving printer-routed information.

PUNCH DEVICE—Device to receive files which contain 2780/3780 punch routing.

ALT PUNCH DEVICE—Alternate device which is automatically selected by the simulator if an error or end of file condition occurs on device receiving punch-routed information.

2. Status Messages

The second area of the VDT display, entitled ****STATUS MESSAGES****, is a list of messages providing a limited historical log of exceptional conditions (see Figure 1-2). The format for a message is:

HM:MM:SS(NN)...message...

| | |
|---|--------------------|
| LINE... | XMT QUEUE FILE.. |
| MODE.. | XMT BATCH... |
| ACTIVE DATA FILE.. | PRINT DEVICE.. |
| ACTIVE DEVICE.. | ALT PRINT DEVICE.. |
| LAST REC NUMBER.. | PUNCH DEVICE.. |
| RCV QUEUE FILE.. | ALT PUNCH DEVICE.. |
| **STATUS MESSAGES** | SIGNON REC... |
| 1) 09:35:26(01) NOT VALID RCV QUEUE | |
| 2) 09:36:20(01) REWIND XMT QUEUE | |
| 3) 09:38:05(63) LINE CONNECTION PENDING | |
| 4) | |
| 5) | |

Figure 1-2 Message Format

Where HH:MM:SS is the system time in hours, minutes, and seconds respectively. NN is a count (01-99) of successive times a message has occurred. This prevents recurrent messages from dominating the message list. Five messages may be displayed simultaneously. Additional messages will then overlay the oldest message screened. **Figure 1-3** is a listing of all status messages.

| | | |
|------------------------------|--|-----------------------------------|
| 00 | | 33 XMT RETRY COUNT EXCEEDED |
| 01 EOF RECEIVED | | 34 |
| 02 EOT RECEIVED | | 35 |
| 03 ILLEGAL PRINT CNTL RCVD | | 36 LOGICAL ENTRY ERROR |
| 04 SWITCHING TO ALTERNATE | | 37 ILLEGAL LOGICAL STATUS |
| 05 OUTPUT DEVICE ERROR | | 38 DATA FORMAT ERROR |
| 06 FATAL ERROR OUTPUT DEVICE | | 40 MNEMONIC CONFLICT |
| 07 RCV RECORD OVERFLOW | | 41 UNABLE TO OPEN PIO |
| 08 CANNOT OPEN RCV FILE | | 42 COMM HARDWARE ERROR |
| 09 INVALID DEVICE IN QUEUE | | 43 ERROR NINETY ONE UNDEFINED |
| 10 CANNOT OPEN XMT FILE | | 44 ILLEGAL MNEMONIC |
| 11 END OF XMT QUEUE | | 45 LIO COMMAND TIMEOUT |
| 12 ESCAPE KEY ABORT | | 46 PIO MODE ERROR |
| 13 RECEIVE ABORT | | 47 COMM ERROR O |
| 14 TRANSMIT ABORT | | 48 |
| 15 COMM ABORT | | 49 |
| 16 RVI ABORT | | 50 DISC READ ERROR |
| 17 XMT QUEUE DELAY | | 51 DISC BUSY |
| 18 CANNOT READ XMT QUEUE | | 52 DISC RECORD SIZE |
| 19 | | 53 DISC NOT READY |
| 20 RVI RCVD | | 54 DISC DATA ERROR |
| 21 LINE CONNECTION PENDING | | 55 DISC ERROR |
| 22 NO COMM DRIVER | | 56 SIGNON ACCEPTED |
| 23 INVALID BID RCVD | | 57 SIGNON SENT, AWAITING RESPONSE |
| 24 NO BID RCVD | | 58 END OF SESSION |
| 25 RCV TIMEOUT | | 59 XMT QUEUE IS CHECKPOINTED |
| 26 SIGNON ERROR | | 60 NO QUEUES DEFINED |
| 27 LINE CONNECTION LOST | | 61 CANNOT OPEN QUEUE |
| 28 NO RESPONSE TO XMT BID | | 62 CANNOT OPEN SIGNON FILE |
| 29 NAK RCVD TO XMT LINE BID | | 63 NOT VALID RCV QUEUE |
| 30 RCV RETRY COUNT EXCEEDED | | 64 NOT VALID XMT QUEUE |
| 31 DISCONNECT RCVD | | 65 REINITIALIZE XMT QUEUE |
| 32 ERROR SENDING EOT | | 66 CANNOT READ SIGNON FILE |

Figure 1-3. Status Messages

3. Simulator Menus

The third area of display consists of various menus and a prompt line. All operator/simulator dialogue is confined to this area. The primary mode selection menu is shown in **Figure 1-1**. Secondary menus are listed in **Figure 1-4** and described in the following paragraphs.

```

LINE...           XMT QUEUE FILE..
MODE..           XMT BATCH...
ACTIVE DATA FILE.. PRINT DEVICE..
ACTIVE DEVICE..  ALT PRINT DEVICE..
LAST REC NUMBER.. PUNCH DEVICE..
RCV QUEUE FILE.. ALT PUNCH DEVICE..
**STATUS MESSAGES** SIGNON REC...

1)
2)
3)
4)
5)

*DEFINE TERMINAL CHARACTERISTICS
1) 2780 STANDARD
2) 2780 WITH MULTIRECORD FEATURE
3) 2780 SINGLE RECORD
4) 3780 STANDARD
5) 3780 EXPANDED TRANSPARENCY

ENTER MODE NUMBER DESIRED ! ■

```

Figure 1-4. Terminal Characteristics

DEFINE TERMINAL CHARACTERISTICS (Figure 1-4)

This mode allows the operator to direct the simulator to use either the 2780 or 3780 protocol and to select various options within these protocols. The simulator options correspond to certain throughput enhancement features available on hard-wired 2780/3780 terminals. It is desirable to select the most efficient option available without exceeding the capabilities existing at the contacted remote terminal or mainframe. Once a mode has been selected by entering the corresponding number 1-5, program control returns to the primary mode selection menu.

2780 STANDARD—This mode will describe to the simulator that only two records per transmission block are to be sent to the remote terminal or host. This is the default mode if no other is chosen.

2780 WITH MULTIRECORD FEATURE—This mode will specify that up to seven records per transmission block are to be sent to the remote terminal or host. This is accomplished by deleting trailing blanks for each record before building a transmission block. This is the most efficient mode for 2780 transmission. If transparent data are being sent, trailing blank deletion will not occur, and this mode becomes equivalent to 2780 STANDARD.

2780 SINGLE RECORD—This mode will specify that only one record per transmission block is to be sent to the remote terminal or host. Single record is the least efficient 2780 mode and should only be used for transmission to non-standard terminals with single record buffers.

3780 STANDARD—This mode will specify that the logical and line level programs are to function in the 3780 protocol. The protocol will be IBM-compatible for a point-to-point network topology. 3780 Standard mode may be used to transmit either text or transparent data. Transparent data transmission in this mode is less efficient than 3780 EXPANDED TRANSPARENCY since the data are sent one record per communications block.

3780 EXPANDED TRANSPARENCY—This mode will enable the 3780 expanded transparency block feature as defined in IBM's component information for the 3780. This mode specifies that six 80-character records will be transmitted in transparent format without record separators. This mode should not be used for text data. This is the recommended mode for Basic/Four to Basic/Four data file transmission.

FILE QUEUE DEFINITION (Figures 1-5 and 1-6)

```

LINE...           XMT QUEUE FILE..
MODE..           XMT BATCH...
ACTIVE DATA FILE.. PRINT DEVICE..
ACTIVE DEVICE..  ALT PRINT DEVICE..
LAST REC NUMBER.. PUNCH DEVICE..
RCV QUEUE FILE.. ALT PUNCH DEVICE..
**STATUS MESSAGES** SIGNON REC...

1)
2)
3)
4)
5)

*QUEUE FILE DEFINITION
PLEASE ENTER THE RECEIVE QUEUE FILE NAME ! ■

```

Figure 1-5. File Queue Definition (Receive)

```

LINE...           XMT QUEUE FILE..
MODE..           XMT BATCH...
ACTIVE DATA FILE.. PRINT DEVICE..
ACTIVE DEVICE..  ALT PRINT DEVICE..
LAST REC NUMBER.. PUNCH DEVICE..
RCV QUEUE FILE.. ALT PUNCH DEVICE..
**STATUS MESSAGES** SIGNON REC...

1)
2)
3)
4)
5)

*QUEUE FILE DEFINITION
PLEASE ENTER THE TRANSMIT QUEUE FILE NAME ! ■

```

Figure 1-6. File Queue Definition (Transmit)

File queues define where data for transmission are found or where data are stored/displayed on reception.

The simulator will first prompt the operator for the file name of the receive file queue:

PLEASE ENTER THE RECEIVE QUEUE FILE NAME!

The operator should enter the name of a receive file queue previously generated by “,QF”. Acceptable file names consist of one to six characters. The file is opened and read. If it is a valid receive file queue, the PRINT DEVICE, ALT PRINT DEVICE, PUNCH DEVICE, ALT PUNCH DEVICE, and RCV QUEUE FILE fields on the VDT are updated and control proceeds to the next prompt. If the file is not available or is not a valid receive queue, the operator is reprompted for the receive file queue name and a status message is displayed (CANNOT OPEN QUEUE or NOT VALID RCV QUEUE). The prompt may be bypassed by entering a carriage return with no file name (Figure 1-7). This simulator will then prompt for the transmit file

queue:

PLEASE ENTER THE TRANSMIT QUEUE FILE NAME!

The operator should enter the name of a transmit file queue previously generated by “,QF”. As above, one to six characters are accepted, and a single carriage return will bypass the prompt. The indicated file is opened, locked, and the first two records are read to verify a valid transmit file queue format. If all is correct the XMT QUEUE FILE field of the VDT is updated and the program returns to the primary mode menu. Otherwise, the operator is reprompted for the transmit file queue name, and a status message is displayed (CANNOT OPEN QUEUE or NOT VALID XMT QUEUE). If the transmit file queue has been checkpointed this will be indicated by a status message (XMT QUEUE IS CHECKPOINTED), since transmission will begin from the checkpoint (see START COMMUNICATIONS section). If the transmit file queue has already been transmitted to completion, the operator is warned (REINITIALIZE XMT QUEUE) and reprompted for the transmit file queue name (Figure 1-8).

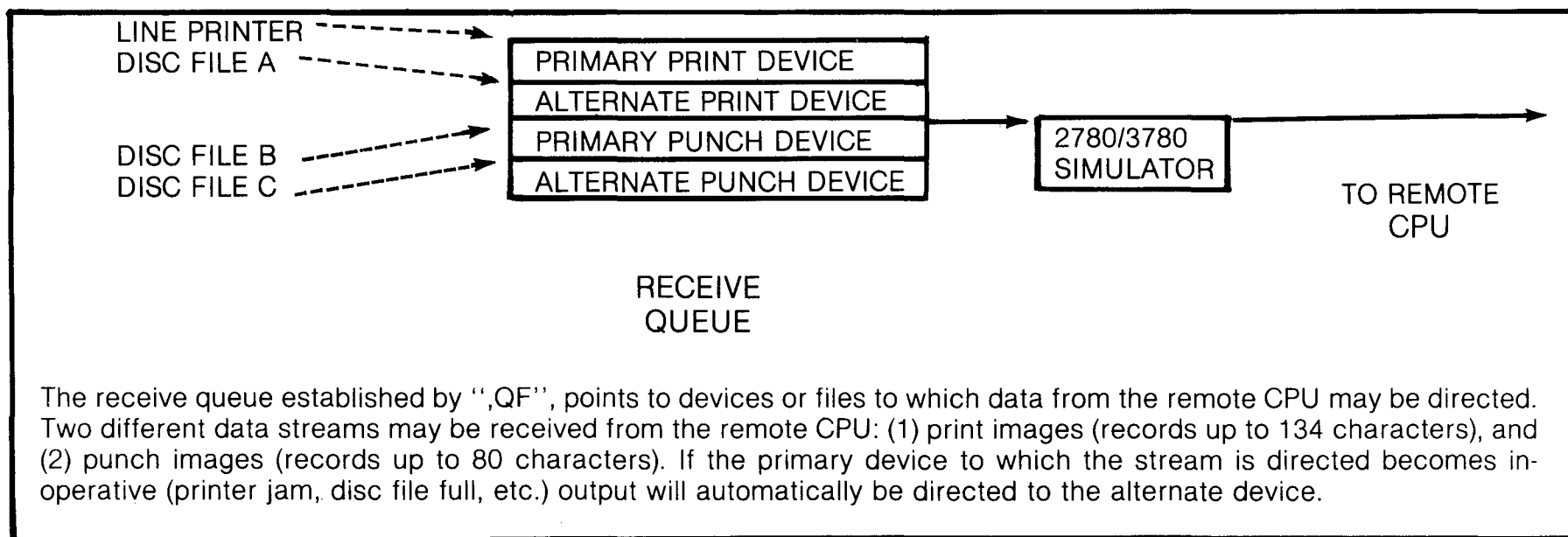


Figure 1-7. Operation of Receive Queue

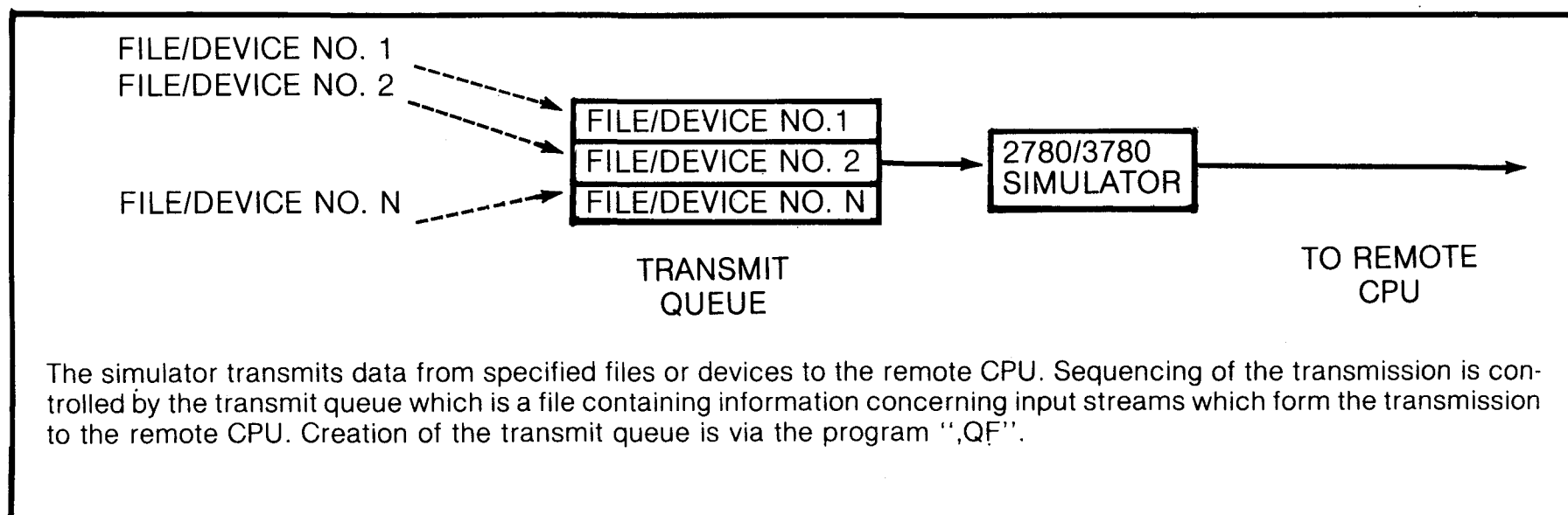


Figure 1-8. Operation of Transmit Queue

The file queue selection made may be entered any number of times to reselect either the transmit or receive file queues. Bypassing either prompt with a carriage return will leave a previously selected queue unchanged.

Signon Definition (Figures 1-9 and 1-10)

```

LINE...          XMT QUEUE FILE..
MODE..          XMT BATCH...
ACTIVE DATA FILE.. PRINT DEVICE..
ACTIVE DEVICE..  ALT PRINT DEVICE..
LAST REC NUMBER.. PUNCH DEVICE..
RCV QUEUE FILE.. ALT PUNCH DEVICE..
**STATUS MESSAGES** SIGNON REC...

1)
2)
3)
4)
5)

*SIGNON MENU
1) /*SIGNON REMOTE 32 PASSWORD WXYZ
2) /*SIGNON
1-714-888-9999 MIS PROCESSOR
3) /*SIGNON REMOTE 6
INVENTORY CONTROL DATA ONLY
CONTINUE SIGNON LIST (Y/N)? ■

```

Figure 1-9. Signon Definition

```

LINE...          XMT QUEUE FILE..
MODE..          XMT BATCH...
ACTIVE DATA FILE.. PRINT DEVICE..
ACTIVE DEVICE..  ALT PRINT DEVICE..
LAST REC NUMBER.. PUNCH DEVICE..
RCV QUEUE FILE.. ALT PUNCH DEVICE..
**STATUS MESSAGES** SIGNON REC...

1)
2)
3)
4)
5)

*SIGNON MENU
4) /*SIGNON REMOTE 16
5) "ABC,789
"838384
COMMENT FOR DOUBLE CARD SIGNON
PLEASE ENTER THE SIGNON NUMBER DESIRED ! ■

```

Figure 1-10. Signon Definition (Double Card)

Remotes communicating with host processors often need to identify themselves after the physical communications link has been established. This is normally done by transmitting a SIGNON card image

to the host. The format and content of the image may vary from host to host, and a particular Basic Four system may communicate to more than one host. Therefore, A SIGNON file has been established and a utility program ("LOGON") provided to build the file.

This mode selection first allows the operator to review the entire SIGNON file. The file is displayed sequentially three records at a time (two records at a time if double card signons are included). After each group is displayed, the operator is prompted:

CONTINUE SIGNON LIST (Y/N)?

Each time the operator responds "Y", the simulator will continue to cycle through the records of the SIGNON file. If the operator responds "N", the program will next prompt for one of the signon images to be chosen by number:

PLEASE ENTER THE SIGNON NUMBER DESIRED!

The first 27 characters of the selected signon will be displayed in the SIGNON REC field of the VDT, and the program will return to the primary mode menu.

The signon selection made may be entered any number of times to reselect the signon. A carriage return in response to the signon number prompt will cause the program to return to the primary mode menu without affecting a previously selected signon. If a signon number greater than any record existing in the SIGNON file is entered, any previously selected signon is "deselected," the SIGNON REC field of the VDT is blanked, and control returns to the primary mode menu.

If for any reason the SIGNON file cannot be accessed, an appropriate status message is displayed (CANNOT OPEN SIGNON FILE or CANNOT READ SIGNON FILE).

Start Communications (Figure 1-11)

```

LINE...          XMT QUEUE FILE..
MODE..          XMT BATCH...
ACTIVE DATA FILE.. PRINT DEVICE..
ACTIVE DEVICE..  ALT PRINT DEVICE..
LAST REC NUMBER.. PUNCH DEVICE..
RCV QUEUE FILE.. ALT PUNCH DEVICE..
**STATUS MESSAGES** SIGNON REC...

1)
2)
3)
4)
5)

*START COMMUNICATIONS

PRIMARY/SECONDARY TRANSMITTER (P/S)? ■

```

Figure 1-11. Start Communications

Once appropriate terminal characteristics, file queues, and signon information have been defined, the operator may start communications. Sessions with a host processor or with another 2780/3780 terminal differ partially in setup requirements and sequence of events to establish and maintain the communications link.

The normal setup for communicating with another 2780/3780 terminal would be to have transmit and receive file queues defined, but no signon defined. Because a signon has not been defined, the simulator will prompt:

PRIMARY/SECONDARY TRANSMITTER (P/S)?

If the operator enters "P", the simulator will proceed to establish the communications link, then enter transmit mode. If the operator enters "S", the simulator will proceed to establish the communications link, then enter receive mode. By prior agreement, one 2780/3780 terminal should transmit first while the other receives first (i.e., in a Basic Four to Basic Four situation, one operator selects "P" while the other selects "S"). If this procedure is not followed, the problem of "contention" arises. Both devices attempt to transmit or receive simultaneously. Communications may or may not be established under these circumstances. Simply entering a carriage return to the primary/secondary prompt will cause the program to return to the primary mode menu without starting communications.

Communications with most mainframe processors require that a signon and a receive file queue have been selected. A transmit file queue will normally also be used, but is optional. Because a signon has been selected, the simulator must by definition be the first side of the communications link to transmit. Thus, the contention problem discussed above is avoided and the program will establish a communications link without displaying the primary/secondary transmitter prompt.

The program will now attempt to establish the communications link. If the system is in a switched or dial-up network, the program will wait for the physical communication link to be established. For a terminal operating in the unattended mode (Auto Answer), the host processor or other 2780/3780 terminal must call to establish the physical link. For a terminal operating in the attended mode, the operator must establish the physical link by dialing the host or other 2780/3780 terminal and making the data set (modem) ready. Until the link is established, the simulator will display the message (LINE CONNECTION PENDING).

If the system is operating on a leased/dedicated network, the physical link is already established, and

communications can begin immediately. In either case, the LINE field of the VDT is now set to OPEN to flag the link as established.

If no signon is selected, the simulator will enter the transmit or receive mode as dictated by the response to the primary/secondary transmitter prompt. Otherwise, the simulator will transmit the selected signon card image(s) as a file, then enter the receive mode to await response from the host processor. This is denoted by a status message (SIGNON SENT, AWAITING RESPONSE). Typical response would be for the host to send log-on information and/or results of previously submitted jobs. If the signon cannot be transmitted, a status message is displayed (SIGNON ERROR) and program control returns to the primary mode menu. If the signon is sent, but the information contained is not acceptable, the host processor will terminate communications.

Once communications has been established, the simulator will alternate between transmit and receive modes until transmit and receive file queues have been exhausted, an unrecoverable error occurs, or the operator manually aborts communications.

Through each cycle in transmit mode, the simulator will transmit one logical file, if available, then relinquish control to the host processor or other remote terminal by sending a 2780/3780 End of Transmission indicator. Access to the data comprising a logical file is controlled by information stored in the transmit file queue: file (or device) names, transmission status, checkpoints, file chains, and data type.

Checkpoints allow the operator to automatically resume an earlier interrupted transmission from the point of interruption. A record checkpoint exists for each file listed in the transmit file queue, and a master file checkpoint exists to mark progress through the transmit queue. When a transmit queue is built, the queue file utility initializes each file checkpoint to the first record of the file and the master file checkpoint to the first file in the queue. The simulator finds the file (or device) name currently indicated by the file checkpoint, opens it (disc files are locked), reads it, and transmits it. The record checkpoint is advanced for each record read. After the file has been transmitted, the file is closed and the file checkpoint is advanced, then both checkpoints are written into the transmit file queue.

Current checkpoint values (corrected for the number of records read, but not yet received at the other side of the communications link) are written into the transmit file queue should an abort occur. A transmit file queue, which has been transmitted to completion, must be "re-wound" before it may be used again for transmission. The queue file utility may be used to reinitialize the record checkpoints and master file checkpoint.

A file history indicator is written into the transmit file queue along with the checkpoints, i.e., "COMPLETED" or "BREAKPOINTED." A history of "BREAKPOINTED" is further marked to differentiate operator intervention versus other errors, i.e., "ESCAPE KEY ABORTED" or "INTERRUPTED."

Each file in the transmit file queue has a transmit status indicator, "READY," "SKIP," or "WAIT." Files marked "READY" are accessed for transmission as previously indicated. Files marked "SKIP" are not transmitted. Instead, the master file checkpoint is advanced to the next file in the queue until a "READY" or "WAIT" file is found. When a file marked "WAIT" is encountered, a status message (XMT QUEUE DELAY) and a prompt are screened:

HOLD/SEND FILE "xxxxxx" (H/S)?

Where xxxxxx is the file (or device) name. If the operator selects "S", the file is sent exactly like a "READY" file. If the operator selects "H", the simulator relinquishes transmit mode by sending a 2780/3780 end of transmission indicator without having sent any data. The checkpoints are not advanced and program control proceeds to the receive mode. The same "WAIT" file will be encountered on every subsequent return to transmit mode until the operator selects "S" (send).

Skip files allow the operator to select which of several optional files listed in the queue to transmit, e.g., alternate JCL files specifying output to printer or to punch. Wait files allow the operator to delay progress through the transmit queue. For instance, one job may not run correctly until a previously submitted job completes. Wait files should not be used in transmit file queues intended for unattended operation.

If the current file in the transmit file queue is chained, the simulator will not revert to receive mode after that file has been transmitted. Rather, the next file in the queue is accessed and transmitted. This continues until an unchained file is found, terminating the chain. A 2780/3780 end of file indicator will be sent only after the last file in the chain. This causes the host processor or other remote terminal to concatenate a series of separate physical files into one logical file. Chaining is convenient for combining files of standardized information with frequently updated information, e.g., adding JCL to program or data files.

Host processors normally support checkpointing only on a complete file basis rather than on a record basis. When communications is started, if a signon was selected (communicating with a host processor), the simulator will rewind the record checkpoint to the start of a checkpointed file. If the checkpoint lies

within a series of chained files, the checkpoint is backed up to the beginning of the chain (start of the **logical** file). If a signon was not selected, the simulator will start transmitting the file from the record indicated in the record checkpoint. Record checkpointing, is of course, only meaningful when applied to random access devices.

After the last entry in the transmit file queue has been transmitted, the transmit file queue is closed and transmit logic is disabled. This is denoted by status messages (END OF XMT QUEUE and TRANSMIT ABORT). Program control passes to receive mode and remains there as long as receive is enabled.

Through every cycle in receive mode, the simulator pauses nominally to allow the host processor or other remote terminal to initiate a transmission. If the host or other remote does not initiate a transmission within that frame time, the simulator reverts to transmit mode. In receive mode, the simulator will accept any number of contiguous files, demarked by 2780/3780 end of file indicators, up to the capacity of the selected receive devices (files). Upon receipt of a 2780/3780 end of transmission indicator, the simulator will automatically revert to transmit mode. Receipt of end of file or end of transmission indicators is denoted by status messages (EOF RCVD or EOT RCVD).

Two receive devices (files) will be opened the first time the simulator enters receive mode and will remain open for the duration of the communications session. These devices (files) are the ones which were specified by the selected receive file queue as PRINT DEVICE and PUNCH DEVICE. Received files will be routed to these (files) according to 2780/3780 routing indicator embedded within the data. Files which contain no routing indicators will default to the PRINT DEVICE. Should an error, including end of file condition, occur on a receive device (file), the simulator will close that device and open the ALT PRINT DEVICE or ALT PUNCH DEVICE, as appropriate, and continue to receive.

The operator may abort communications at any time by depressing the ESCAPE key twice. If the abort occurs while the communications LINE is OPEN, the simulator will take necessary steps to close active devices (files), checkpoint the transmit file queue, and close the communications line. This is indicated by status messages (TRANSMIT ABORT, RECEIVE ABORT, and COMM ABORT). An additional status message (ESCAPE KEY ABORT) is screened to differentiate this abort from ones generated by internal errors. If the ESCAPE key is depressed twice while communications is not active, the simulator will exit to console mode.

| | | |
|-------------------|--|-----------------------|
| PAGE 10 | TITLE 2780/3780 SIMULATOR SYSTEM REFERENCE MANUAL SECTION 1 2780/3780 SIMULATOR | DATE 8/1/78 |
|-------------------|--|-----------------------|

Some modems may not terminate the physical communications link after the simulator has closed the line. In this case, the operator must "hang-up" the phone to prevent a large phone bill.

SET DATE/TIME (Figure 1-12)

| | |
|---------------------|--------------------|
| LINE... | XMT QUEUE FILE.. |
| MODE.. | XMT BATCH... |
| ACTIVE DATA FILE.. | PRINT DEVICE.. |
| ACTIVE DEVICE.. | ALT PRINT DEVICE.. |
| LAST REC NUMBER.. | PUNCH DEVICE.. |
| RCV QUEUE FILE.. | ALT PUNCH DEVICE.. |
| **STATUS MESSAGES** | SIGNON REC... |
| 1) | |
| 2) | |
| 3) | |
| 4) | |
| 5) | |
| *SET DATE/TIME | |
| ENTER MONTH (1-12) | |
| ENTER DAY (1-31) | |
| ENTER YEAR (0-99) | |
| ENTER HOUR (0-23) | |
| ENTER MIN (0-59) | |
| ENTER SEC (0-59) ■ | |

Figure 1-12. Set Date/Time

This mode selection allows the operator to set the system date (MONTH, DAY, YEAR) and the system time (HOUR, MONTH, SECOND) for logging purposes.

USER DATA FILE INTERFACE

Successful and efficient operation of the 2780/3780 simulator depends greatly upon proper definition of user data files and appropriate description of these files in the transmit and receive file queues.

Systems that receive data from a 2780 expect 80-byte card image records. To comply with this requirement, the simulator will deblock data file records read into 80-byte card image records before transmission. Deblocking has no apparent effect on files whose records are 80 bytes long. Files with records longer or shorter than 80 bytes will appear "skewed" after transmission. Deblocking of files whose total size (record length x number of records) is not an integral multiple of 80, produces a short record when the last card image is formed. This residual record is padded to a full 80 bytes using blanks in text mode or nulls in transparent mode.

Host processors frequently attach positional significance to data in JCL or program files relative to an 80-byte card image. The contents of "pure" data files are irrelevant to the host processor. Skewed records would not prevent successful transmission; however, the data would need to be restored to its original format for later use at the host processor.

Thus "pure" data files for transmission to a host processor may be created to contain other than 80 bytes of data. An application program on the host processor could be used to reformat the data, if it were skewed in transmission.

Data used by a 2780/3780 is classified as text (character) data or transparent (binary) data. Each file in the transmit file queue is marked for the simulator as text or transparent. The content of data received by the simulator determines whether it is text or transparent.

Systems receiving data from a 2780/3780 expect text data to be in the EBCDIC character set. All files which are marked in the transmit file queue as text will be translated from ASCII to EBCDIC prior to transmission. (see Table 1).

Conversely, all text data received by the simulator will be translated from EBCDIC to ASCII (see Table 2).

Transparent data may assume any values and so are not restricted to any particular character set. All files which are marked transparent in the transmit file queue will be sent as read, without translation. Transparent data received will not be translated.

Chained files in the transmit file queue represent a single logical file. Thus all files in a chain must be marked uniformly either as text data or as transparent data. If transparent data must be chained to JCL and/or program files for communications with a host processor, the JCL and/or program files should be created as EBCDIC data and marked transparent in the transmit file queue.

Non-Standard Record Length

The simulator can reformat data received in transparent mode. Records will be reblocked from the 80-byte card images received to the record size specified by the receive file queue. Thus, records longer or shorter than 80 bytes may be transmitted Basic/Four to Basic/Four without any apparent skewing.

To transfer long or short records successfully, the transmitting side of the link must mark the file transparent in its transmit file queue. The receiving side of the link must have the record size field of the selected receive file queue set equal to the actual record size of the file to be transferred.

Routing and Print Control

A 2780/3780 terminal is required to perform receive device routing and printer carriage control based on special character sequences embedded within received data. These characters are stripped from the data when the function is performed. Files received from a host processor will contain routing and/or print

Table 1 ASCII to EBCDIC Conversion

| B8-B5 \ B4-B1 | | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
|---------------|---|-------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|----------|-----------|-----------|----------|----------|----------|-----------|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| 1000 | 8 | NULL 00 | SOH 01 | STX 02 | ETX 03 | EDT 37 | ENQ 2D | ACK 2E | BEL 2F | BS 16 | HT 05 | LF 25 | VT OB | FF OC | CR OD | SO OE | SI OF |
| 1001 | 9 | DLE 10 | DC1 11 | DC2 12 | DC3 13 | DC4 3C | NAK 3D | SYN 32 | ETB 26 | CAN 18 | EM 19 | SUB 3F | ESC 27 | FS 1C | GS 1D | RS 1E | US 1F |
| 1010 | A | SPACE 40 | ! 5A | " 7F | # 7B | \$ 5B | % 6C | & 50 | /' 7D | (4D |) 5D | * 5C | + 4E | , 6B | _ 60 | · 4B | / 61 |
| 1011 | B | 0 FO | 1 F1 | 2 F2 | 3 F3 | 4 F4 | 5 F5 | 6 F6 | 7 F7 | 8 F8 | 9 F9 | : 7A | ; 5E | < 4C | = 7E | > 6E | ? 6F |
| 1100 | C | @ 7C | A C1 | B C2 | C C3 | D C4 | E C5 | F C6 | G C7 | H C8 | I C9 | J D1 | K D2 | L D3 | M D4 | N D5 | O D6 |
| 1101 | D | P D7 | Q D8 | R D9 | S E2 | T E3 | U E4 | V E5 | W E6 | X E7 | Y E8 | Z E9 | [CO | \ EO | [DO | ↑ 5F | ← 6D |
| 1110 | E | ‘ 79 | a 81 | b 82 | c 83 | d 84 | e 85 | f 86 | g 87 | h 88 | i 89 | j 91 | k 92 | l 93 | m 94 | n 95 | o 96 |
| 1111 | F | p 97 | q 98 | r 99 | s A2 | t A3 | u A4 | v A5 | w A6 | x A7 | y A8 | z A9 | { CO | 6A | } DO | ~ A1 | DEL 07 |

Table 2 EBCDIC to ASCII Conversion

| B8-B5 | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | | 0 | 80 | 81 | 82 | 83 | BF | 89 | BF | FF | BF | BF | BF | 8B | 8C | 8D | 8E |
| 1 | 90 | 91 | 92 | 93 | BF | BF | 88 | BF | 98 | 99 | BF | BF | 9C | 9D | 9E | 9F | |
| 2 | BF | BF | BF | BF | BF | 8A | 97 | 9B | BF | BF | BF | BF | BF | BF | 85 | 86 | 87 |
| 3 | BF | BF | 96 | BF | BF | BF | BF | 84 | BF | BF | BF | BF | 94 | 95 | BF | 9A | |
| 4 | AO | BF | BF | BF | BF | BF | BF | BF | BF | BF | BF | AE | BC | A8 | AB | FC | |
| 5 | A6 | BF | BF | BF | BF | BF | BF | BF | BF | BF | BF | A1 | A4 | AA | A9 | BB | DE |
| 6 | AD | AF | BF | BF | BF | BF | BF | BF | BF | BF | BF | FC | AC | A5 | DF | BE | BF |
| 7 | BF | BF | BF | BF | BF | BF | BF | BF | BF | BF | EO | BA | A3 | CO | A7 | BD | A2 |
| 8 | BF | E1 | E2 | E3 | E4 | E5 | E6 | E7 | E8 | E9 | BF | BF | BF | BF | BF | BF | BF |
| 9 | BF | EA | EB | EC | ED | EE | EF | EO | F1 | F2 | BF | BF | BF | BF | BF | BF | BF |
| A | BF | EE | F3 | F4 | F5 | F6 | F7 | F8 | F9 | FA | BF | BF | BF | BF | BF | BF | BF |
| B | BF | BF | BF | BF | BF | BF | BF | BF | BF | BF | BF | BF | BF | BF | BF | BF | BF |
| C | DB | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | BF | BF | BF | BF | BF | BF | BF |
| D | DD | CA | CB | CC | CD | CE | CF | DO | D1 | D2 | BF | BF | BF | BF | BF | BF | BF |
| E | DC | BF | D3 | D4 | D5 | D6 | D7 | D8 | D9 | DA | BF | BF | BF | BF | BF | BF | BF |
| F | BO | B1 | B2 | B3 | B4 | B5 | B6 | B7 | B8 | B9 | BF | BF | BF | BF | BF | BF | BF |

control based on JCL and programs executed at the host processor. A 2780/3780 terminal may transmit files containing routing and/or print control to another 2780/3780 but never to a host processor.

Files to be transmitted by the simulator may include routing and print control sequences. These special character sequences cannot usually be keyed directly into the data but must instead be created by a BASIC program.

A punch routing sequence may appear only in the first character position(s), left most characters, of the first record of a logical file. (First file only of a chain in the transmit file queue). 2780 punch routing is "ESC" "4", 1BB4 hex in ASCII or 27F4 hex in EBCDIC. Do not use the hex value 9B for an ASCII ESC. 3780 punch routing is "DC1", 91 hex in ASCII or II hex in EBCDIC.

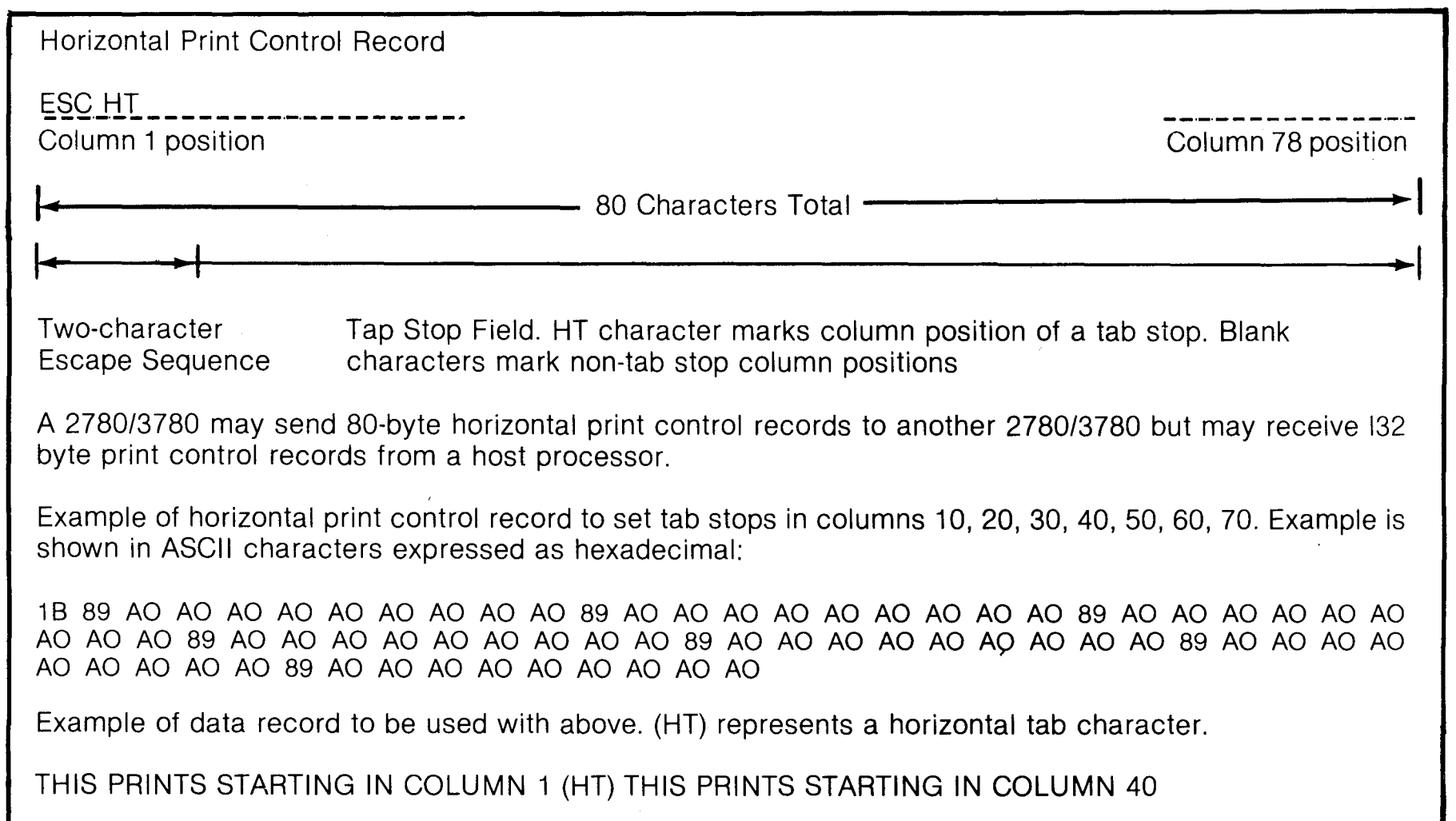
Printer control may include both horizontal and vertical carriage control functions. Horizontal print control provides for spacing between fields on a print line. A special non-data horizontal print control record is sent to define tab stops. On encountering an "HT" character in succeeding data records, the simulator will insert blanks into the record from the position of the HT until the relative position of the next tab stop in the horizontal print control record. The result is similar to depressing a tabulator key on a typewriter, causing the carriage to space over to the next tab stop.

Table 4 Vertical Print Control

| FUNCTION | TWO-CHARACTER SEQUENCE |
|---------------------------|------------------------|
| FORM FEED | ESC A |
| SLEW TO CHANNEL 2 | ESC B |
| SLEW TO CHANNEL 3 | ESC C |
| SLEW TO CHANNEL 4 | ESC D |
| SLEW TO CHANNEL 5 | ESC E |
| SLEW TO CHANNEL 6 | ESC F |
| SLEW TO CHANNEL 7 | ESC G |
| SLEW TO CHANNEL 8 | ESC H |
| *SLEW TO CHANNEL 9 | ESC I |
| *SLEW TO CHANNEL 10 | ESC J |
| *SLEW TO CHANNEL 11 | ESC K |
| *SLEW TO CHANNEL 12 | ESC L |
| *OVERPRINT | ESC M |
| OVERPRINT | ESC ± |
| SINGLE SPACE | ESC / |
| DOUBLE SPACE | ESC S |
| TRIPLE SPACE | C T |
| *3780 ONLY | |

NOTE: When ESC is coded in Business BASIC 8-Bit ASCII, use 1B hex. E.G. 20 LET A\$ = \$1Bc1\$ + A\$ would add ESC A to the character string A\$.

Table 3 Horizontal Print Control



Any number of print control records may be sent within a file to redefine the tab stops. If a transmission is restarted from a checkpoint, the operator must ensure that the record checkpoint is backed up to include the last horizontal tab control record, otherwise the results would be unpredictable.

Vertical print control provides for spacing between print lines. These control characters may appear on any record in a file but must always be the leftmost characters. 2780/3780 print control characters are converted to BASIC compatible print control mnemonics. The record is always printed before the print control function is performed.

Use of these facilities increases transmission speed by eliminating the need to transmit fields of blanks or blank lines to effect output formatting. See **Tables 3 and 4** for details of horizontal and vertical print control.

Transmission Devices

A 2780/3780 is basically a unit record device with fixed record lengths and sequentially accessed files. Although the simulator will function using variable length records or random access files, best results are obtained using files which approximate the unit record format.

Disc files are read using BASIC statements READ or READRECORD respectively, depending on whether the file is marked text or transparent in the transmit file queue. Differences between READ/WRITE and READRECORD/WRITERECORD dictate a number of limitations for the use of disc files:

- Files originally created with WRITERECORD must be read using READRECORD.
- Pure binary files must be written using WRITERECORD.
- Accessing un-initialized records with READ may cause disc errors.
- Multiple field records should be written using WRITERECORD. If WRITE is used, only the first field will be transmitted.
- Files, created with variable length records using WRITE can be read using READRECORD to force a fixed record length (BASIC will pad the record with nulls to the defined record length). This method may be used to transmit BASIC programs in LIST format.

All records of a disc file will be transmitted, including what may be unused records at the end of the file. JCL and program files must be defined so as to contain no unused (un-initialized) records. Such garbage records may cause a host processor to terminate communications.

Serial files will not work in general, since the variable length records will become impossibly

scrambled in transmission due to record deblocking. Direct files will not work in general, since sequential access of a randomly created file will access un-initialized records.

Key Record

Keystation files are created on line by the operator. Host processor systems like HASP, ASP, or POWER define commands which may be entered from an RJE terminal. Keystation files are ideal for this purpose, allowing the operator to query the host system about status of his jobs, to start or terminate functions, send messages, etc. When a keystation file is encountered in the transmit file queue, the operator is prompted.

The operator may enter up to 80 characters terminated by a carriage return. The simulator will blank fill records to 80 characters. The carriage return is not included as part of the data. Entering a carriage return without any data will generate an end of file. At that point, the simulator will advance the master file checkpoint to the next entry in the transmit file queue and revert to receive mode. If a motor bar key is entered without data, the EOF is issued and the simulator reverts to receive mode, but the master file checkpoint is not advanced. The keystation file will be retained as the top of the transmit file queue until a file is closed with a carriage return rather than a motor bar. This allows the operator to dynamically control the number of keystation files issued. Typically, a one-line command would be transmitted as a single file. While the simulator was in receive mode, the host processor would evaluate the command and send back its response.

If the first character of the first record keyed in the file is "CNTRL," (Control Key and Comma simultaneously), the simulator will automatically put punch routing on the file. Punch routing may be transmitted to another remote terminal, but never to a host processor.

Receive Devices

Records received from a host processor will contain up to 80 bytes of data if punch routed, and up to 134 bytes (132 data and 2 print control) if printer routed. Records received from another 2780/3780 terminal will be up to 80 bytes in length. The simulator may reformat received data to other lengths in transparent mode.

Disc files should be indexed files with defined record length greater than or equal to the record sizes to be received. All records are written to disc using the BASIC statement WRITERECORD. 2780/3780 printer carriage control will be written to disc verbatim. Any later offline printing must provide for appropriate handling of the print control.

A single disc file may spool multiple received files

| | | |
|-------------------|--|-----------------------|
| PAGE 14 | TITLE 2780/3780 SIMULATOR SYSTEM REFERENCE MANUAL SECTION 1 2780/3780 SIMULATOR | DATE 8/1/78 |
|-------------------|--|-----------------------|

in the following manner. When a 2780/3780 EOF indicator is received, the existing disc file is segmented into two files. The first file bears the original file name and consists of exactly those records received. The second file is given a new name, as described below, and contains all the yet unused records of the original file. The new file name is written into the receive file queue so that during subsequent communication sessions, the simulator will resume reception to the next available disc file. This process may continue until all the available record space originally allocated is consumed. Reception of any further records will generate an error causing the simulator to switch to the alternate device. The alternate device could, of course be another disc file.

Disc file names are limited to the form XXXNNN where XXX is any three alphacharacters and NNN is a numeric value 000-999. New file names are created in the file segmenting process described by adding 1 to the numeric portion of a name, e.g., "ABCO23" would become "ABCO24." If the simulator attempts to create a new file using a name which is already defined, a critical error will occur.

NOTE

Any disc drives which are not ready must be DISABLED during a communications session, receiving to disc. Otherwise, a critical error will occur, and the last file received will be lost.

Line Printer—Records in excess of 132 bytes are truncated on the right to 132 bytes, and a status message (RCV RECORD OVERFLOW) is displayed. 2780/3780 printer carriage control will be stripped from the data and the required function performed, if possible. Print control functions which may not be performed within BASIC are marked by a status message (ILLEGAL PRINT CNTL RCVD). Binary data should not be routed to the printer.

VDT—Records are displayed single spaced. 2780/3780 print control will be stripped from the records and no function performed. Records in excess of 80 bytes will overflow to succeeding lines.

Files routed to a VDT other than the one controlling the simulator will use the entire display.

A VDT which has already been STARTed, other than the VDT controlling the simulator, may not be used as a receive device. Binary data should not be routed to a VDT.

SOFTWARE STRUCTURE

The simulator is structured at three levels. This scheme separates the functional, logical, and physical processing as one system task (i.e., one TCB) and interface with a separate task, the physical I/O processor through a common dictionary.

The functional, or operational, level is written in the BASIC language. The tasks performed at this level include operator interface, file handling, and upper-level communications flow control. This level consists of two overlays which perform off-line and on-line communications functions. This level is not protocol dependent and may be used in the development of similar communications products.

The second level, or logical I/O, processor is written in assembly language and is the intermediary between the functional processing and the physical line handler. The tasks performed at this level include physical/logical record handling, status reporting to the functional level, data transfer between physical and functional processors, and provides an intermediate control code response handling. The structure is shared by both the 2780 and 3780 protocol. Certain protocol dependent (2780 or 3780) subroutines are contained within the logical level.

The physical, or line level, processor is written in assembly language and is a separate task. The functions performed at this level include the necessary logic for physical control of the communications equipment (modem adapter) and the logical sequence of events on that line (transmit/receive data, ACK, NAK). Generally, it accepts/passes physical data blocks and status to/from the logical I/O processor through I/O buffer queues and executes the necessary line protocol sequence to initiate the transfer between the terminal and the remote.

SECTION 2

QUEUE FILE UTILITY

PROGRAM FUNCTION

The queue file utility generates and modifies files that are used by the 2780/3780 Simulator program. These files are used by the simulator as queues for the reception and transmission of data.

Program “*O” must be used to define and allocate the appropriate disk storage for each queue file. The queue files must be INDEXED with a record length of 40 bytes. The number of records in a transmit file is variable, depending on the user’s need. Receive queue files contain three records.

OPERATING CHARACTERISTICS

The following paragraphs describe the operation of the utility and the interaction between the program and the operator.

The utility is activated by the operator by keying in RUN “,QF”. The program will then prompt the operator for the necessary responses to build/modify a transmit or a receive queue.

General Inquiry/Response Sequences

The program will prompt the operator for the mode of operation desired (see Figure 2-1).

```
QUEUE FILE UTILITY

ENTER NUMBER OF OPERATION DESIRED (CR
TO EXIT):
1. GENERATE NEW QUEUE.
2. ADD ENTRY TO TRANSMIT QUEUE
3. MODIFY TRANSMIT QUEUE ENTRY
4. DELETE ENTRY FROM TRANSMIT QUEUE.
5. MODIFY RECEIVE QUEUE.
6. DISPLAY QUEUE.
7. INITIALIZE TRANSMIT QUEUE
```

Figure 2-1. Program Mode Selection Menu

If a carriage return is keyed, the program will exit. Otherwise, the program will enter one of the modes specified by the initial prompt.

Generate New Queue

The program will prompt the operator for the name of the file to be operated on.

ENTER QUEUE FILE NAME:

The operator should enter the name of the file to be generated. The name is limited to a six-character alphanumeric sequence.

The program will prompt the operator for the type of file to be created.

ENTER QUEUE TYPE NUMBER:
1 X RECEIVE
2 X TRANSMIT

The operator should enter the number of queue file type to be generated.

The program will prompt the operator to enter the queue I.D. Up to 6 characters may be entered. This field is not required and may be left blank.

ENTER QUEUE I.D.:

```
QUEUE FILE UTILITY - GENERATE NEW QUEUE

ENTER QUEUE FILE NAME:
ENTER QUEUE TYPE: TRANSMIT
ENTER QUEUE I.D.:

ENTRY nnn
ENTER DEVICE TYPE:
ENTER DEVICE UNIT NUMBER:
ENTER FILE NAME (CR TO EXIT):
ENTER FILE TYPE:
ENTER INITIAL TRANSMIT STATUS:
SHOULD THIS FILE BE CHAINED TO NEXT
ENTRY (Y/N)?
```

Figure 2-2. Generate Transmit Queue

Generate Transmit Queue (See Figure 2-2)

If the queue type selected is transmit, the program will proceed through the necessary inquiry/response sequences necessary to build each entry.

The next prompt will be to obtain the device type. The operator may specify one of several device types to be used for output.

ENTER DEVICE TYPE NUMBER (CR TO EXIT):
1 X DISC
2 X MAG TAPE
3 X CONSOLE

The operator should enter the appropriate number associated with the desired device type. The program will insert the proper code in the DEVICE TYPE field for the current queue record. A carriage return (CR) entered at this point will indicate end of file. The new queue will be written on the disc and the program will return to the operation desired mode (Figure 2-1).

The next prompt is to enter the unit number of the device selected. This is essential for multi-unit devices such as magnetic tape drives. It will not be necessary to specify a disc drive number since the system will be able to find a particular file from its file name.

ENTER UNIT NUMBER:

The program will accept a number from 0-99 and enter this number in the device number field of the queue record.

The next prompt will be to enter the file name which contains the data to be transmitted.

ENTER FILE NAME:

For disc files, the program will accept a one-to-six-character alphanumeric sequence for the file name. Two-character system device names will automatically be generated for magnetic tape and console based on the previously entered device type and Device Unit Number (e.g., Terminal #3, "T3").

The next prompt determines the type of data to be transmitted. If the data set contains data other than ASCII alphanumeric characters, the operator must inform the simulator that the data must be transmitted in TRANSPARENT mode.

ENTER DATA TYPE NUMBER:

- 1 x ASCII
- 2 x TRNSP

The next prompt determines initial transmit file status. This is a flag to the simulator which "enables" this defined data set or device to be used. The choices given will enable the file to be used immediately (ready) or will indicate that the simulator should pause or wait before continuing with the transmit sequence.

The program will prompt the operator for the initial status of the data file with the following message:

ENTER INITIAL TRANSMIT STATUS:

- 1 x READY
- 2 x HOLD
- 3 x SKIP

The operator should enter a 1 to define the queue

entry as ready, a 2 to define the queue entry as in the hold state, or a 3 to indicate this entry should be skipped.

The utility will enter the appropriate code in the TRANSMIT STATUS field of the record being built.

The next prompt sets the chain parameter. This parameter determines whether or not the logical file defined for transmit will span multiple physical files in the queue.

Any logical file may consist of multiple physical disc files or a mixture of disc and tape files. This allows the simulator to transmit a sequence of JCL (Job Control Language) from a disc file to be concatenated with other physical records from various devices.

SHOULD THIS FILE BE CHAINED TO NEXT ENTRY (Y/N)?

If the operator responds with "Y" (YES), the next physical file is concatenated to the logical file.

If the operator responds with "N" (NO), the transmit file defined is the end of a logical file and is not concatenated with another.

The program will now write the record it has been compiling and return to Generate Transmit Queue (Figure 2-2) to allow the operator to add another record or end the file.

Generate Receive Queue (See Figure 2-3)

| | |
|---|-------------------|
| QUEUE FILE UTILITY - GENERATE NEW QUEUE | |
| ENTER QUEUE FILE NAME: | |
| ENTER QUEUE TYPE: RECEIVE | |
| ENTER QUEUE I.D.: | |
| PRINT RECORD SIZE: | |
| PRIMARY PRINT DEVICE: | UNIT NUMBER: |
| PRIMARY PRINT FILE NAME: | |
| ALTERNATE PRINT DEVICE: | UNIT NUMBER: |
| ALTERNATE PRINT FILE NAME: | |
| PUNCH RECORD SIZE | |
| PRIMARY PUNCH DEVICE: | UNIT NUMBER: |
| PRIMARY PUNCH FILE NAME | |
| ALTERNATE PUNCH DEVICE: | UNIT NUMBER: |
| ALTERNATE PUNCH FILE NAME | |
| DEVICE NUMBERS: | |
| 1 = DISC | 4 = PRINTER #N |
| 2 = PRINTER #1 | 5 = MAGNETIC TAPE |
| 3 = CONSOLE | |

Figure 2-3. Generate Receive Queue

If the queue was defined as a receive queue, the program will build the receive queue file records.

Files that are received with a printer designation are routed according to the Print File information in the receive queue file. This information includes the record length (usually 132 characters), the primary print device, unit number, and file name; and the alternate print device, unit number, and file name.

Normally, any print files received will be routed to the primary print device. If that device is not available, the file will be sent to the alternate print device. For example, assume that the primary print device is Printer No. 1 and the alternate is the disc, file name "RCV000". Received files with a printer routing would be sent to printer No. 1 unless it was busy or out of order. In that case, the files would be sent to "RCV000" on the disc and could be printed later.

Also included in the receive queue file is routing information for punch data. The operator must enter the primary punch device, unit number, and file name; and the alternate punch device, unit number, and file names.

The devices available for receive files are disc, printer No. 1, printers No. 2 through N, console, and magnetic tape.

NOTE

All receive disc file names must be in the format AAANN: three alphabetic characters followed by three numeric characters. Two character system device names will automatically be generated for Console, Printer No. 2 through N, and Magnetic Tape based on the previously entered device type and device unit number. Printer No. 1 is a special case and will generate both unit number and two-character device name when device type is selected.

Add Entry To Transmit Queue (See Figure 2-4)

This phase of the program allows the operator to

```

QUEUE FILE UTILITY - ADD ENTRY TO
TRANSMIT QUEUE

ENTER QUEUE FILE NAME:
QUEUE I.D.:

ENTRY nnn
ENTER DEVICE TYPE:
ENTER DEVICE UNIT NUMBER:
ENTER FILE NAME (CR TO EXIT):
ENTER DATA TYPE:
ENTER INITIAL TRANSMIT STATUS:
SHOULD THIS FILE BE CHAINED TO NEXT
ENTRY (Y/N)?

```

Figure 2-4. Add Entry to Transmit Queue

add one or more entries at the end of an existing transmit queue. New entries may only be added at the end.

The program will prompt the operator to enter the transmit queue file name desired. The name can be up to six characters long. If a carriage return is entered, the program will return to the program mode selection menu (**Figure 2-1**).

ENTER QUEUE FILE NAME:

If the queue selected is not a transmit queue, or does not exist, an error message will be displayed. Otherwise the queue I.D. will be displayed.

Modify Transmit Queue Entry

This phase of the program allows the operator to change existing records in a transmit queue file.

The program will prompt the operator to enter the name of the queue field to be modified.

ENTER QUEUE FILE NAME:

Up to six characters may be entered. If a carriage return is entered, the program will return to the program mode selection menu (**Figure 2-1**).

If the queue file selected is not a transmit queue, an error message will be displayed. Otherwise, the queue I.D. will be displayed.

The program will prompt the operator to enter the number of the next file in the queue to be transmitted. If a carriage return is entered, the current value will be left unchanged.

The program will read each entry in the queue and display the entry number and the data file name. Each entry will be numbered (**See Figure 2-5**).

```

QUEUE FILE UTILITY - MODIFY TRANSMIT
QUEUE ENTRY

ENTER QUEUE FILE NAME:
QUEUE I.D.:
NEXT ENTRY TO BE TRANSMITTED:

ENTRY #      FILE NAME
  1          xxxxxx
  2
  3
  :
  :
  n
ENTER ENTRY NUMBER TO BE MODIFIED
(CR TO EXIT):

```

Figure 2-5. Entry Number and File Name (Modify)

The program will prompt the operator to define which of the records is to be modified.

ENTER ENTRY NUMBER TO BE MODIFIED:

The program will read the selected record and display its contents. If a carriage return is entered as a response, the program will return to the program mode selection menu (Figure 2-1). Each item displayed will be numbered (see Figure 2-6).

```

QUEUE FILE UTILITY - MODIFY TRANSMIT
QUEUE ENTRY

ENTER QUEUE FILE NAME:
QUEUE I.D.:

ENTRY nnn

1. FILE NAME:
2. TRANSMIT STATUS:
3. DEVICE TYPE:
4. DEVICE UNIT NUMBER:
5. FILE CHAIN:
6. DATA TYPE:
7. LAST PHYSICAL RECORD:
8. LAST LOGICAL RECORD:
9. FILE HISTORY:

ENTER NUMBER OF ITEM TO BE MODIFIED:
(CR TO EXIT):

```

Figure 2-6 Selected Record Contents (Modify)

The program will prompt the operator for the item number he desires to modify.

ENTER NUMBER OF ITEM TO BE MODIFIED:

The operator will then be prompted to enter the new value for the field selected. If a carriage return is entered instead of an item number, the program will return to ENTER QUEUE FILE NAME to allow another record to be modified.

Each time an item in an entry is modified, the display of that record will be updated to reflect its current contents.

Delete Entry From Transmit Queue

This phase of the program allows the operator to delete an entry from a transmit queue file.

The program will prompt the operator to enter the name of the queue file.

ENTER QUEUE FILE NAME:

Up to six characters may be entered. If a carriage return is entered the program will return to the program mode selection menu (Figure 2-1).

If the queue file selected is not a transmit queue, an error message will be displayed. Otherwise, the queue I.D. will be displayed.

The program will read each entry in the queue and display the entry number and the data file name. Each entry will be numbered (see Figure 2-7).

```

QUEUE FILE UTILITY - DELETE ENTRY
FROM TRANSMIT QUEUE

ENTER QUEUE FILE NAME:
QUEUE I.D.:

ENTRY #      FILE NAME
1            xxxxxx
3
.
.
.
n

ENTER ENTRY NUMBER TO BE DELETED
(CR TO EXIT):

```

Figure 2-7. Entry Number and File Name (Delete)

The program will prompt the operator to enter the number of the record to be deleted.

ENTER ENTRY NUMBER TO BE DELETED:

If a carriage return is entered, the program will return to program mode selection (Figure 2-1).

If an entry is selected, it will be displayed and the operator will be asked to confirm that this is the record to be deleted (see Figure 2-8).

```

QUEUE FILE UTILITY - DELETE ENTRY
FROM TRANSMIT QUEUE.

ENTER QUEUE FILE NAME:
QUEUE I.D.:

ENTRY nnnn

1. FILE NAME:
2. TRANSMIT STATUS:
3. DEVICE TYPE:
4. DEVICE UNIT NUMBER:
5. FILE CHAIN:
6. DATA TYPE:
7. NUMBER OF PHYSICAL RECORDS
TRANSFERRED:
8. NUMBER OF LOGICAL RECORDS
TRANSFERRED:
9. FILE HISTORY:

ENTER NUMBER OF ITEM TO BE DELETED:

```

Figure 2-8 Selected Record Contents (Delete)

IS THIS THE ENTRY TO BE DELETED (Y/N)?

The record will only be deleted if a "Y" is entered in response to this question. If the record is deleted, all following records will be moved up one position in the file. In either case, the program will re-display the contents of the file and then return to ENTER ENTRY NUMBER TO BE DELETED to allow another record to be deleted.

queue file I.D. will be displayed.

The program will display the file entries with numbers and prompt the operator to:

ENTER NUMBER OF ITEM TO BE MODIFIED:

If a carriage return is entered in response, modification will be ended and the program will return

```

                                QUEUE FILE UTILITY - MODIFY RECEIVE QUEUE

                                ENTER QUEUE FILE NAME:
                                QUEUE I.D.:

                                1. PRINT RECORD SIZE:
                                2. PRIMARY PRINT DEVICE:
                                3. UNIT NUMBER:
                                4. PRIMARY PRINT FILE NAME:
                                5. ALTERNATE PRINT DEVICE:
                                6. UNIT NUMBER:
                                7. ALTERNATE PRINT FILE NAME:
                                8. PUNCH RECORD SIZE:
                                9. PRIMARY PUNCH DEVICE:
                                10. UNIT NUMBER:
                                11. PRIMARY PUNCH FILE NAME
                                12. ALTERNATE PUNCH DEVICE:
                                13. UNIT NUMBER:
                                14. ALTERNATE PUNCH FILE NAME:

                                ENTER NUMBER OF ITEM TO BE MODIFIED (CR TO EXIT):
                                DEVICE TYPE NUMBERS:
                                1 = DISC
                                2 = PRINTER #1
                                3 = CONSOLE
                                4 = PRINTER #N
                                5 = MAG TAPE

```

Figure 2-9. Modify Receive Queue

If the last record in the file is deleted, the file will be erased and the operator will be informed.

Modify Receive Queue

This phase of the queue file utility program allows the operator to change parameters in an existing receive queue (see Figure 2-9).

The program will prompt the operator to enter the name of the receive queue to be modified.

ENTER QUEUE FILE NAME:

Up to six characters may be entered. If a carriage return is entered, the program will return to the program mode selection menu (Figure 2-1).

If the queue file selected is not a receive queue, an error message will be displayed. Otherwise, the

to program mode selection (Figure 2-1). If a number is entered, the program will prompt the operator to enter the new value for the item selected and then re-display the file with the new item.

Display Queue

This phase of the program allows the operator to display a receive or transmit queue on the VDT or the printer.

The program will prompt the operator to select the display device:

DISPLAY ON PRINTER OR VDT (P/V)?

The file will then be displayed on the chosen medium.

If a transmit file is being displayed on the VDT, there may be more records than can be displayed on

a single screen. In that case, when the screen is full, the program will stop and wait for the operator to enter a carriage return before continuing.

(Figure 2-10 is a transmit queue display and Figure 2-11 is a receive queue display).

restore a transmit queue to its initial values after it has been transmitted. It will clear the NEXT ENTRY TO BE TRANSMITTED, LAST PHYSICAL RECORD, LAST LOGICAL RECORD, and FILE HISTORY fields. This allows the queue to be re-used. File names and device parameters will remain as originally defined.

Initialize Transmit Queue

This phase of the program allows the operator to

```

QUEUE FILE UTILITY - DISPLAY QUEUE

ENTER QUEUE FILE NAME:
DISPLAY ON PRINTER OR VDT (P/V)?
NUMBER OF ENTRIES IN QUEUE:
QUEUE TYPE: TRANSMIT NEXT ENTRY TO BE TRANSMITTED:
QUEUE I.D.:

ENTRY   FILE      FILE      LAST PHYS.      LAST LOG DEVICE  UNIT CHAIN DATA  FILE
NO.     NAME      STATUS    RECORD          RECORD          UNIT  CHAIN  TYPE  HISTORY
      xxxx   xxxxxxxx  xxxxxxxx.   9999            9999  xxxxxx  99   999  xxxxx  xxxxxxxxxx

      .
      .
      .

('CR TO EXIT
  
```

Figure 2-10. Transmit Queue Display

```

QUEUE FILE UTILITY - DISPLAY QUEUE

ENTER QUEUE FILE NAME:
DISPLAY ON PRINTER OR VDT (P/V)?
NUMBER OF ENTRIES IN QUEUE: 2
QUEUE TYPE: RECEIVE
QUEUE I.D.:

PRINT RECORD LENGTH:
PRIMARY PRINT DEVICE:          UNIT NUMBER:
PRIMARY PRINT FILE NAME:      UNIT NUMBER:
ALTERNATE PRINT DEVICE:      UNIT NUMBER:
ALTERNATE PRINT FILE NAME
PUNCH RECORD LENGTH:
PRIMARY PUNCH DEVICE:          UNIT NUMBER:
PRIMARY PUNCH FILE NAME:      UNIT NUMBER:
ALTERNATE PUNCH DEVICE:      UNIT NUMBER:
ALTERNATE PUNCH FILE NAME:

('CR TO EXIT)
  
```

Figure 2-11. Receive Queue Display

SECTION 3 SIGNON FILE UTILITY

PROGRAM FUNCTION

This utility is used to generate and modify the ,SIGNON file which is a collection of signon card images and comments which are accessed by the 2780/3780 Simulator. The utility is used to specify the content and number of signon images available for a particular installation.

Signon cards contain information required to establish communications between a remote 2780/3780 terminal and a host mainframe. A remote terminal initiates the signon procedure by transmitting a file consisting of one or sometimes two signon records. The host mainframe scans the transmitted file and continues or terminates communication based on the correctness of the signon information.

OPERATING CHARACTERISTICS

The utility is activated by the operator by keying in RUN ",LOGON." The program will prompt the operator to determine the mode: generate new file, add to an existing file, modify an existing file, or delete from an existing file. The prompting message will read as shown in **Figure 3-1**.

```

1) GENERATE NEW FILE
2) DISPLAY OR ADD TO EXISTING FILE
3) MODIFY EXISTING ENTRY
4) DELETE EXISTING ENTRY
ENTER FUNCTION NUMBER: █
```

Figure 3-1. Function Menu

Generate New File

Generating a new file consists of clearing the residing "number of records" field in the first record of the file ,SIGNON and then entering the ADD TO EXISTING FILE logic of the utility. The ,SIGNON file is defined by the system at configuration time.

Add to Existing File (See Figure 3-2)

Adding records to an existing file consists of a

```

1)
/*SIGNON      REMOTE 5
1-877-0896    PARTS LIST FILE ONLY

2)
/*SIGNON      REMOTE 22 PASSWORD ZZXC

3)
"ABC,123      GGK
"7273
922-4603     DOUBLE CARD SIGNON
ADD NEW RECORD (Y/N)?
```

Figure 3-2. Add To Existing File

series of operator inquiry/responses via the VDT to obtain signon card and remark images which are to be written into the ,SIGNON file.

The program displays the current content of the ,SIGNON file with each entry enumerated. Following each signon entry will be the remark record associated with that signon card.

After displaying the file, the program will prompt for a new signon to be added:

ADD NEW RECORD (Y/N)?

When "Y" is entered, the operator may key in a signon image. The program will then add the completed image to the end of the file and update the number of signon records. If the maximum defined file size of ,SIGNON has not been reached, the program will then display the new contents of the file and prompt the operator for the next signon to be added.

If the operator attempts to enter additional signon images after the maximum defined file size of ,SIGNON has been reached, the program will return to the function menu defined in Figure 3-1 and display the message:

FILE FULL!

When "N" is entered in response to the "ADD NEW RECORD" prompt, the program returns to the function menu.

Modify Existing Entry (See Figure 3-3)

The user is provided a means to correct a signon entry.

The program will read and display the contents of the ,SIGNON file with each entry enumerated.

```

1)
/*SIGNON      REMOTE 5
1-877-0896    PARTS LIST FILE ONLY

2)
/*SIGNON      REMOTE 22 PASSWORD ZZXC

3)
"ABC,123      GGK
"7273
992-4603     DOUBLE CARD SIGNON

ENTER NUMBER OF SIGNON TO BE
MODIFIED █
```

Figure 3-3. Modify Existing Entry

The operator will be prompted with the following message:

ENTER NUMBER OF SIGNON TO BE MODIFIED

The operator should enter the number corresponding to the displayed signon message to be modified. If a RETURN is keyed, the program will return to the original prompt as defined in Figure 3-1.

The selected signon will be displayed to allow modification. When all segments of the signon entry have been modified, the program will display the new contents of the file and prompt the operator again for the next entry to be modified.

Delete Existing Entry (See Figure 3-4)

The user is provided a means to delete a signon entry. The program will read and display the contents of the ,SIGNON file with each entry enumerated.

The operator will be prompted with the following message:

ENTER NUMBER OF SIGNON TO BE DELETED

The operator should enter the number corresponding to the displayed signon message to be deleted. If a RETURN is keyed, the program will return to the original prompt as defined in Figure 3-1.

```

1)
/*SIGNON      REMOTE 5
1-877-0896    PARTS LIST FILE ONLY

2)
/*SIGNON      REMOTE 22 PASSWORD ZZXC

3)
"ABC,123      GGK
"7273
992-4603      DOUBLE CARD SIGNON

ENTER NUMBER OF SIGNON TO BE
DELETED █

```

Figure 3-4. Delete Existing Entry

The selected signon will be deleted by moving up all following signons in the file and decrementing the signon record count. The program will then display the new contents of the file and prompt the operator again for the next entry to be deleted.

Keying in Signon Images (See Figures 3-5 and 3-6)

When the operator is required to key a signon im-

age, the following is displayed:

ENTER SIGNON CARD IMAGE

On the first line, the operator keys in up to 80 characters of a signon card image followed by a CR. Then the program will prompt:

DOUBLE CARD SIGNON (Y/N)?

If "Y" is entered, a flag is set to indicate that the record will contain a double card signon, and the operator is presented with a second 80-character card image to key.

```

                                     23
/*SIGNON      REMOTE 10

DOUBLE CARD SIGNON (Y/N)? █

```

Figure 3-5. Signon Image Display

If "N" is entered, the flag is set to indicate that the second signon card image will not be used. The second signon line will not be displayed and thus remains transparent to the operator.

In either case, the program will now prompt:

TELEPHONE NUMBER (Y/N)?

```

"ABC,123      GGK
   6
"7273

TELEPHONE NUMBER (Y/N)? █

```

Figure 3-6. Signon Image

If "Y" is entered the next line must be input as a phone number field followed by a comment field. Only digits 1 through 9 and a minus (-) character, used to separate digit sequences, are allowed in the phone number field. The program will accept up to 14 characters in this field, which is the maximum necessary for a long—distance call outside the user's area code.

EXAMPLE:
 1-305-727-5555 Long distance, another
 area code
 1-727-5555 Long distance, same area
 code
 727-5555 Local call

The phone number field is terminated by entering a blank or carriage return. (If any comments are desired, the operator must delimit the phone number with at least one blank). A total of 80 characters, including phone number and comments, is allowed on this line.

If "N" is entered, the operator may key this line as a freeform 80-character field.

The position of the next character to be entered will be bracketed above by a column count (1-80) and below by a cursor. MOTOR BARS I and II will be used

to forward and backward space the cursor to the desired position for correction within a field. The cursor will not, however, backspace to a previous field or forward space to a field not yet entered. A carriage return will be used to signal the end of input to a field. The program will either advance to the next field to be entered or terminate the edit session after the last field.

SYSTEM IMPACT

Disc Storage

The ,SIGNON file requires the necessary storage for an indexed file with two or more 241-byte records. For example, if five signon card images are required, then space for six records of 241 bytes is required.

The signon file is included in the customer disc at configuration time. The current procedure is to reserve space for six records, or 1,446 bytes.

SECTION 4

JOB CONTROL LANGUAGE UTILITY

PROGRAM FUNCTION

The Job Control Language (JCL) utility provides file management of 80-character CARD IMAGE RECORDS which are typically used to transmit job control language from a Basic Four computer system to a mainframe computer. In most cases these JCL records are concatenated to data files to build the entire stream submitted to a mainframe.

OPERATING CHARACTERISTICS

The following paragraphs describe operation of the utility and interaction between the program and the operator.

The JCL utility is activated by the operator by keying in RUN "UJCL." The program then prompts the operator for responses to perform necessary management of JCL files.

General Inquiry/Response Sequences

The program first prompts the operator with the message:

ENTER JCL FILE NAME:

If only a carriage return is entered, the program exits. Otherwise, after the operator enters the file name, the following message is displayed:

ENTER JCL TYPE ('A' = ASCII/'E' = EBCDIC):

The JCL type entered here should match the one specified in the queue file, (i.e., 'A' for ASCII text; 'E' for EBCDIC text).

If a JCL file has previously been defined as one type, but the other type is entered, an error message INCORRECT JCL TYPE is displayed and the operator is reprompted to enter a correct type.

If the file is not currently defined on disc, the program prompts:

FILE file name HAS NOT BEEN ALLOCATED
DO YOU NEED FILE file name TO BE ALLOCATED
(Y/N)?

If the operator responds with an "N", the program will start again. If a "Y" is entered, the following message will be displayed:

ENTER DISC NUMBER FOR FILE ALLOCATION:

The operator then responds with the proper disc number.

The program then requests:

ENTER NUMBER OF RECORDS:

The operator then responds with the number of records to be included in the file.

After the number of record is entered, the program starts file allocation and initialization. If completion is normal, the following will be displayed:

FILE file name IS ALLOCATED AT sector
number
ENTER 'CR' TO CONTINUE:

If completions is abnormal, one of the following will be displayed:

DISC ERROR: nn

or

DISC DIRECTORY IS FULL

If completion is normal, the selective menu will then be displayed:

ENTER NUMBER OF FUNCTION:

1. DISPLAY RECORDS
2. GENERATE RECORDS
3. MODIFY RECORDS
4. ADD RECORDS
5. DELETE RECORDS

The operator now enters the desired choice and is further prompted by the system, as described in the following paragraphs.

1. Display Records

If the DISPLAY RECORDS function is selected, the following message will be displayed:

ENTER STARTING RECORD NUMBER:

Up to five records from the file, starting with the one specified, will be displayed on the lower half of the screen at a given time:

RECORD n
(contents of record n)
RECORD n + 1
(contents of record n + 1)
.
.
.
.
RECORD n + 4
(contents of record n + 4)
ENTER 'CR' TO CONTINUE

If 'CR' is entered, up to five additional records are displayed.

If the end of file is encountered, the program displays:

END OF FILE
ENTER 'CR' TO CONTINUE

| | | |
|-----------------------|---|-------------------|
| DATE 8/1/78 | TITLE 2780/3780 SIMULATOR SYSTEM REFERENCE MANUAL SECTION 4 JOB CONTROL LANGUAGE UTILITY | PAGE 27 |
|-----------------------|---|-------------------|

If CR' is entered after this prompt, the program will return to the selective menu.

2. Generate Records

If this function is selected, the following message will be displayed:

ENTER STARTING RECORD NUMBER

After the record number is entered, the column number indicators will be displayed as follows:

RECORD n

1 2 3 4 5 6 7 8
123456789012345678901234567890123456789012345678901234567890

Each character entered will move the cursor one digit to the right. A backspace will not move the cursor backward.

Note

Motor Bars III and IV are used to move the cursor forward and backward, respectively, without altering the original character in the column skipped.

If a 'CR' is entered, the cursor will not move through 80 columns and the record will be written blank-filled to disc. The record number is incremented and the cursor will return to column 1. Motor Bar I also returns the cursor to column 1 without writing the record to disc. Motor Bar II will cause the last record to be written to disc and the program will then return to the selective menu.

3. Modify Records

If this function is selected, the following message will be displayed:

ENTER STARTING RECORD NUMBER

After the record number is entered, the column number indicators will be displayed as follows:

RECORD n

1 2 3 4 5 6 7 8
123456789012345678901234567890123456789012345678901234567890

The contents of the record will be displayed under the column indicators and the cursor will be positioned under the first character. Motor Bar functions are identical to those in GENERATE RECORDS

4. Add Records

This function is provided to expand a pre-defined JCL file which is too small to contain all JCL records needed. If this function is selected, the following messages will be displayed:

FILE file name CONTAINS n RECORDS
ENTER NUMBER OF RECORDS TO BE
ADDED

After the number of records desired are entered, the program will allocate a new file of the size desired, transfer existing data from the old file to the new one, erase the old file, and re-name the new file with the old file name. The program will then enter the GENERATE RECORDS mode.

5. Delete Records

This function tailors a pre-defined JCL file which is too large. If this function is selected, the following message will be displayed:

ENTER STARTING RECORD NUMBER TO BE
DELETED:

After the number of record is entered, all records from that point to the end of file will be deleted. When this function is completed, the program will return to the selective menu.

REFERENCES

IBM Component Description: IBM 2780 Data Transmission Terminal, GA17-3005-3

IBM Component Information for the IBM 3780 Data Communications Terminal, GA17-3063-3

IBM General Information—Binary Synchronous Communications, GA27-3004-1

IBM Synchronous Data Link Control General Information

BELL SYSTEM Data Set 201C Interface Specification, PUB41210

BELL SYSTEM Catalog of Technical References, PUB 400000

Data Pro Reports on Data Communications Data Pro Research, Delran, N.J. (609) 764-0100

IBM System/360 Operating System Job Control Language, C28-6539-9

IBM System/360 Operating System Concepts and Facilities, C28-6535-4

IBM System/360 Operating System Data Management Services, GC26-3746

| | | |
|-----------------------|---|-------------------|
| DATE 8/1/78 | TITLE 2780/3780 SIMULATOR SYSTEM REFERENCE MANUAL REFERENCES | PAGE 29 |
|-----------------------|---|-------------------|

APPENDIX

This appendix describes the various steps and procedures that are undertaken at a terminal to fully setup and execute the simulator program.

Figure 1

The program “,UJCL” is used to build Job Control Language (JCL) files of 80-character records which become part of a job stream to the mainframe.

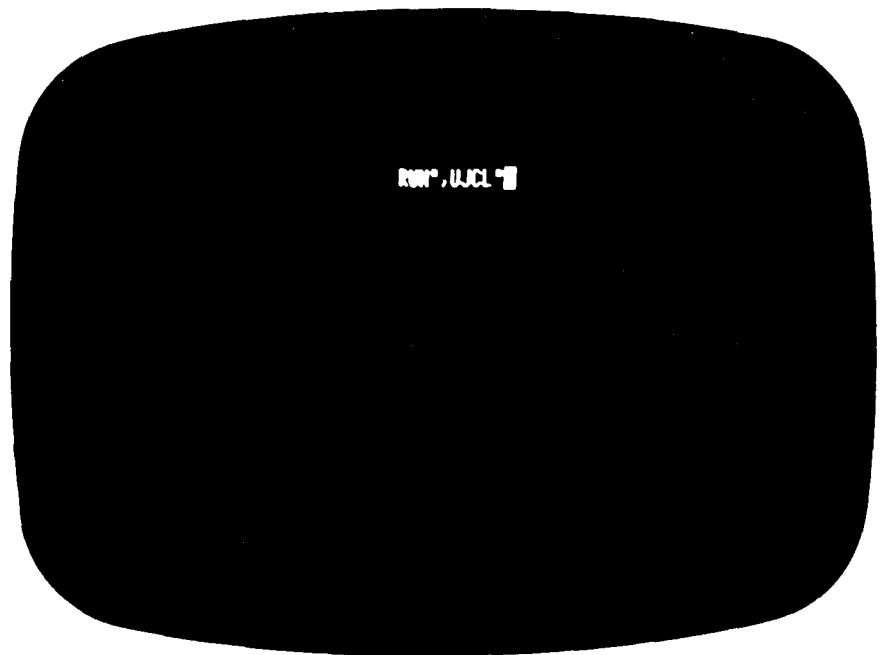


Figure 2

After JCL files are built, the program “*O” is utilized to create queue files required for execution of the 2780/3780 emulation program.

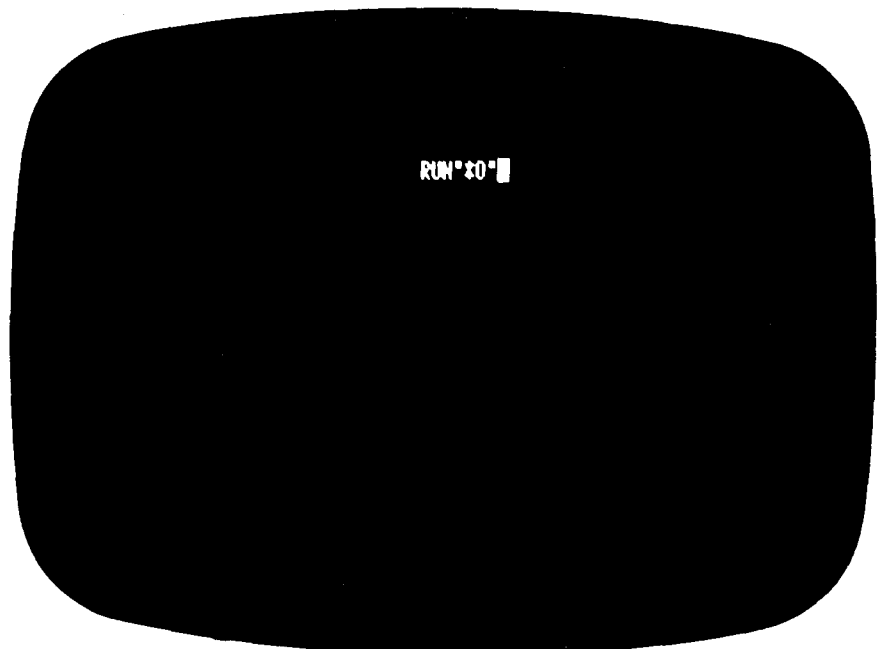
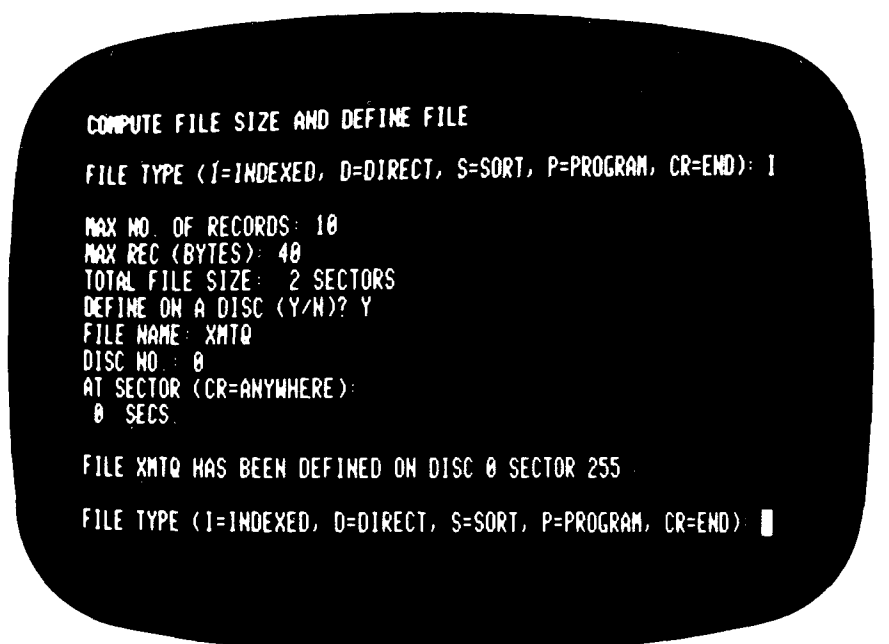


Figure 3

The transmit queue file is allocated using “*O”. All transmit queue files must be indexed with each record 40 bytes long. The number of records in the transmit queue must be at least one greater than the number of files to be transmitted to the receiving station. Receive queue files are also created using the program “*O”. All receive queue files must be indexed, record length 40 bytes, and three records long.



```
RUN OF 1
```

Figure 4

The queue file utility program “,QF” is used to create queues for simulator input and output. The queues indicate devices which receive data from the remote system or are repositories of data for transmission to the remote system.

```
QUEUE FILE UTILITY
```

```
ENTER NUMBER OF OPERATION DESIRED (CR TO EXIT): 1
```

1. GENERATE NEW QUEUE
2. ADD ENTRY TO TRANSMIT QUEUE
3. MODIFY TRANSMIT QUEUE ENTRY
4. DELETE ENTRY FROM TRANSMIT QUEUE
5. MODIFY RECEIVE QUEUE
6. DISPLAY QUEUE
7. INITIALIZE TRANSMIT QUEUE

Figure 5

Execution of the queue file utility begins with the display of a primary mode selection with seven options providing facilities for creation and maintenance of queue files. In this case, option 1 “GENERATE NEW QUEUE” is selected.

```
QUEUE FILE UTILITY - GENERATE NEW QUEUE  
ENTER QUEUE FILE NAME (CR TO EXIT): XMTQ
```

```
ENTER QUEUE TYPE NUMBER: TRANSMIT  
ENTER QUEUE I.D.: TEST
```

```
ENTRY 1  
ENTER DEVICE TYPE NUMBER (CR TO EXIT): DISC  
ENTER UNIT NUMBER: 0  
ENTER FILE NAME: JCL1  
ENTER DATA TYPE NUMBER: ASCII  
ENTER INITIAL TRANSMIT STATUS: READY  
SHOULD THIS FILE BE CHAINED TO NEXT ENTRY (Y/N)? Y
```

Figure 6

This display shows the transmit queue “XMTQ” being established. The file “JCL1” which was created with “,UJCL” is established as the first part of a logical file for transmission.

Figure 7

This display illustrates a second file, "DATA" which is established as the second part of the logical file for transmission.

```
QUEUE FILE UTILITY - GENERATE NEW QUEUE
ENTER QUEUE FILE NAME (CR TO EXIT): XMTQ

ENTER QUEUE TYPE NUMBER: TRANSMIT
ENTER QUEUE I.D.: TEST

ENTRY 2
ENTER DEVICE TYPE NUMBER (CR TO EXIT): DISC
ENTER UNIT NUMBER: 0
ENTER FILE NAME: DATA
ENTER DATA TYPE NUMBER: ASCII
ENTER INITIAL TRANSMIT STATUS: READY
SHOULD THIS FILE BE CHAINED TO NEXT ENTRY (Y/N)? Y
```

Figure 8

A third file "JCL2" is chained to the first two files ("JCL1" and "DATA"). Since this is the last part of the logical file, the user specifies that it is not chained to another entry. This display completes creation of the chained transmit file.

```
QUEUE FILE UTILITY - GENERATE NEW QUEUE
ENTER QUEUE FILE NAME (CR TO EXIT): XMTQ

ENTER QUEUE TYPE NUMBER: TRANSMIT
ENTER QUEUE I.D.: TEST

ENTRY 3
ENTER DEVICE TYPE NUMBER (CR TO EXIT): DISC
ENTER UNIT NUMBER: 0
ENTER FILE NAME: JCL2
ENTER DATA TYPE NUMBER: ASCII
ENTER INITIAL TRANSMIT STATUS: READY
SHOULD THIS FILE BE CHAINED TO NEXT ENTRY (Y/N)? N
```

Figure 9

This display illustrates the receive queue being established. Primary print output is directed to printer No. 1, while the secondary output (if the printer becomes not ready for any reason) is directed to the disc file RCV000. Punch output will be directed to file RCV100; secondary punch output will be directed to file RCV200.

```
QUEUE FILE UTILITY - GENERATE NEW QUEUE
ENTER QUEUE FILE NAME (CR TO EXIT): RCVQ

ENTER QUEUE TYPE NUMBER: RECEIVE
ENTER QUEUE I.D.: TEST

PRINT RECORD SIZE: 132
PRIMARY PRINT DEVICE NUMBER: PRINTER #1      UNIT NUMBER: N/A
PRIMARY PRINT FILE NAME: LP                    UNIT NUMBER: N/A
ALTERNATE PRINT DEVICE NUMBER: DISC           UNIT NUMBER: N/A
ALTERNATE PRINT FILE NAME: RCV000
PUNCH RECORD SIZE: 80
PRIMARY PUNCH DEVICE NUMBER: DISC             UNIT NUMBER: N/A
PRIMARY PUNCH FILE NAME: RCV100
ALTERNATE PUNCH DEVICE NUMBER: DISC           UNIT NUMBER: N/A
ALTERNATE PUNCH FILE NAME: RCV200

DEVICE TYPE NUMBERS:
1 = DISC
2 = PRINTER #1
3 = CONSOLE
4 = PRINTER #N
5 = MAG TAPE
```

```
QUEUE FILE UTILITY - DISPLAY QUEUE
ENTER QUEUE FILE NAME (CR TO EXIT): XMTQ
```

```
DISPLAY ON PRINTER OR VDT (P/V) ? U
NUMBER OF ENTRIES IN QUEUE: 3
QUEUE TYPE: TRANSMIT      NEXT ENTRY TO BE TRANSMITTED: 1
QUEUE I.D.: TEST
```

| ENTRY NO. | FILE NAME | FILE STATUS | LAST PHYS RECORD | LAST LOG RECORD | DEVICE | UNIT NO. | CHAIN | DATA TYPE | FILE HISTORY |
|-----------|-----------|-------------|------------------|-----------------|--------|----------|-------|-----------|--------------|
| 1 | JCL1 | READY | 0 | 0 | DISC | 0 | 2 | ASCII | NO HIST. |
| 2 | DATA | READY | 0 | 0 | DISC | 0 | 3 | ASCII | NO HIST. |
| 3 | JCL2 | READY | 0 | 0 | DISC | 0 | 0 | ASCII | NO HIST. |

```
(CR TO EXIT)
```

Figure 10

The transmit queue "XMTQ" is displayed with the three files "JCL1", "DATA," and "JCL2" concatenated to form the job stream transmitted to the remote system.

```
QUEUE FILE UTILITY - DISPLAY QUEUE
ENTER QUEUE FILE NAME (CR TO EXIT): RCVQ
```

```
DISPLAY ON PRINTER OR VDT (P/V) ? U
NUMBER OF ENTRIES IN QUEUE: 2
QUEUE TYPE: RECEIVE
QUEUE I.D.: TEST
```

| | |
|-----------------------------------|----------------|
| PRINT RECORD SIZE: 132 | UNIT NUMBER: 0 |
| PRIMARY PRINT DEVICE: PRINTER #1 | UNIT NUMBER: 0 |
| PRIMARY PRINT FILE NAME: LP | UNIT NUMBER: 0 |
| ALTERNATE PRINT DEVICE: DISC | UNIT NUMBER: 0 |
| ALTERNATE PRINT FILE NAME: RCV000 | UNIT NUMBER: 0 |
| PUNCH RECORD SIZE: 80 | UNIT NUMBER: 0 |
| PRIMARY PUNCH DEVICE: DISC | UNIT NUMBER: 0 |
| PRIMARY PUNCH FILE NAME: RCV100 | UNIT NUMBER: 0 |
| ALTERNATE PUNCH DEVICE: DISC | UNIT NUMBER: 0 |
| ALTERNATE PUNCH FILE NAME: RCV200 | UNIT NUMBER: 0 |

```
(CR TO EXIT)
```

Figure 11

The receive queue "RCVQ" is displayed.

```
READY
>RUN, LOGON
```

Figure 12

The program ",LOGON" is executed to create signon records.

Figure 13

The primary mode selection menu for “,LOGON” is displayed. Options are provided for creation and maintenance of signon record image files.



Figure 14

The signon image displayed on the screen is stored on disc. Disc-based signon files consist of 80-character blank-filled records.

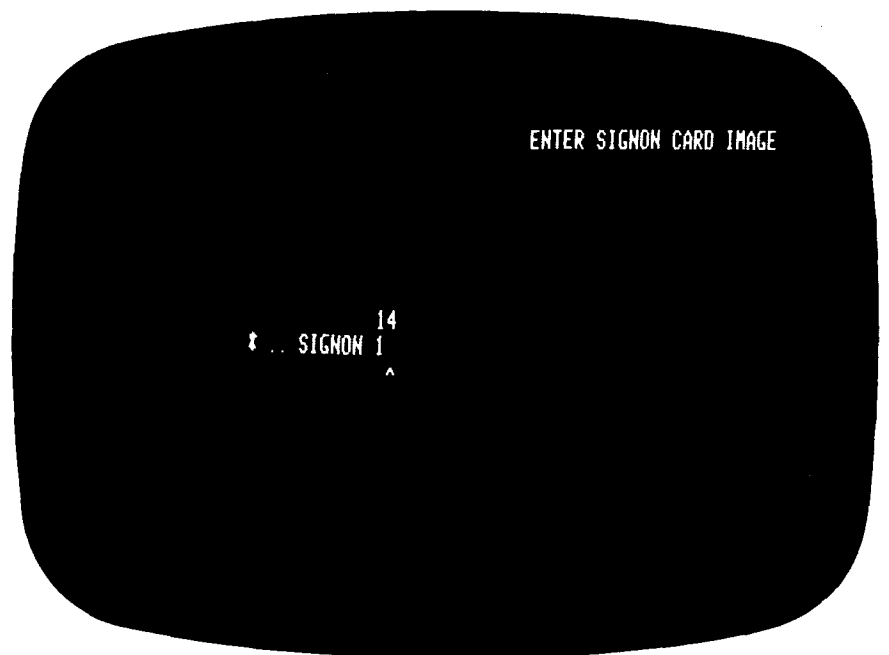


Figure 15

The user is asked whether a double sign card image is desired. Generally, this response is negative. The user may next enter a telephone number and comments, which are used solely for information purposes.



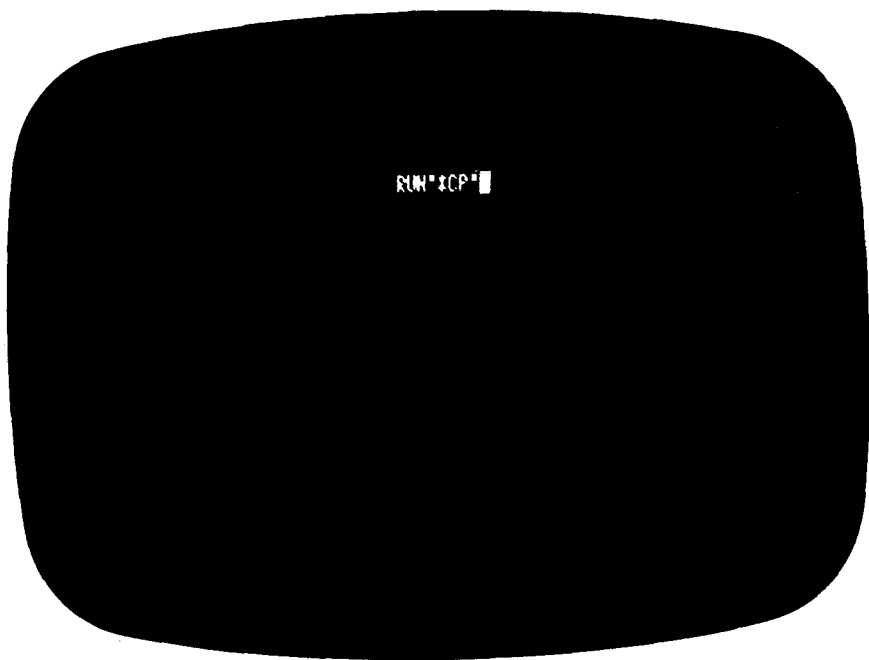


Figure 16

The user executes the 2780/3780 simulator named "*CP".

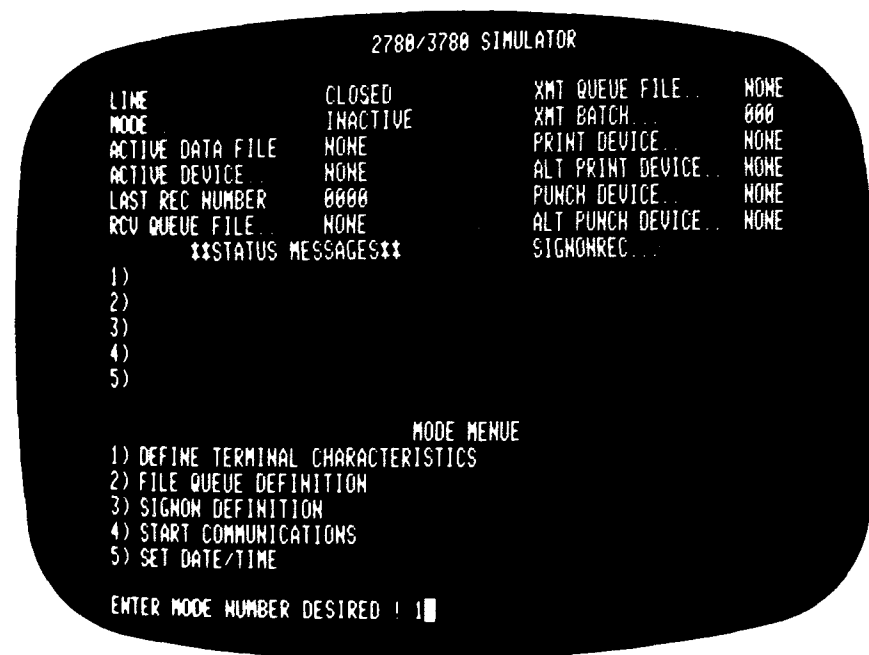


Figure 17

The program "*CP" displays a three-part screen including: (1) real-time operational status of the simulator, (2) status messages, which is a historical log of exception conditions that have occurred during simulator operation, and (3) mode menus and prompt line.

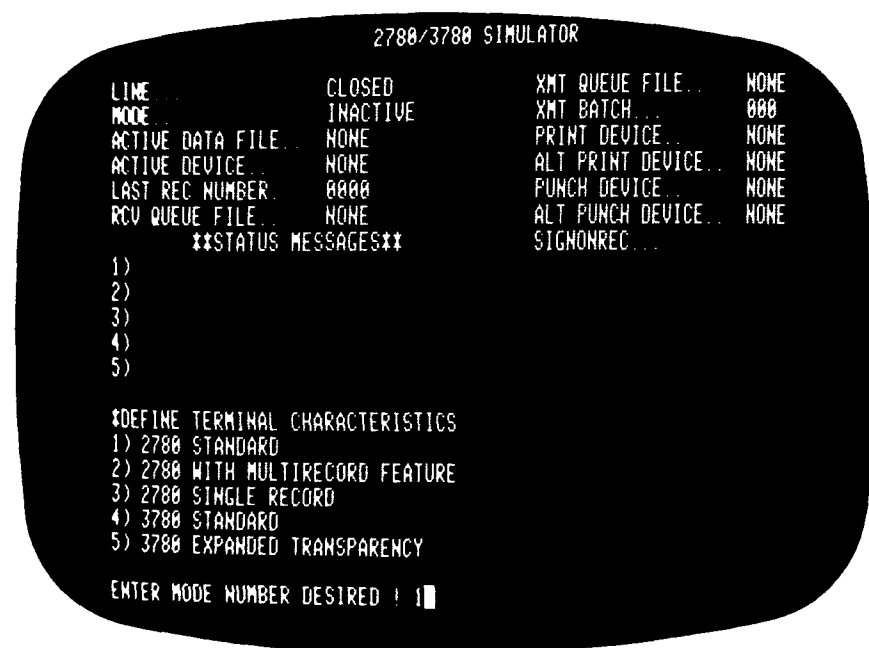


Figure 18

There are five separate types of terminal characteristics that may be selected. In this particular case, 2780 standard is selected.

Figure 19

The primary mode selection menu is displayed. The user is prompted to specify the queues used in the transmission session with the remote computer.

```
2780/3780 SIMULATOR
LINE...      CLOSED      XMT QUEUE FILE.. NONE
MODE...      INACTIVE    XMT BATCH...     000
ACTIVE DATA FILE.. NONE    PRINT DEVICE...  NONE
ACTIVE DEVICE..  NONE    ALT PRINT DEVICE.. NONE
LAST REC NUMBER.. 0000    PUNCH DEVICE...  NONE
RCV QUEUE FILE..  NONE    ALT PUNCH DEVICE.. NONE
                **STATUS MESSAGES**
                SIGNONREC...

1)
2)
3)
4)
5)

                                MODE MENU
1) DEFINE TERMINAL CHARACTERISTICS
2) FILE QUEUE DEFINITION
3) SIGNON DEFINITION
4) START COMMUNICATIONS
5) SET DATE/TIME

ENTER MODE NUMBER DESIRED | 2
```

Figure 20

The simulator requests the name of the receive queue file: "RCVQ".

```
2780/3780 SIMULATOR
LINE...      CLOSED      XMT QUEUE FILE.. NONE
MODE...      INACTIVE    XMT BATCH...     000
ACTIVE DATA FILE.. NONE    PRINT DEVICE...  NONE
ACTIVE DEVICE..  NONE    ALT PRINT DEVICE.. NONE
LAST REC NUMBER.. 0000    PUNCH DEVICE...  NONE
RCV QUEUE FILE..  NONE    ALT PUNCH DEVICE.. NONE
                **STATUS MESSAGES**
                SIGNONREC...

1)
2)
3)
4)
5)

*QUEUE FILE DEFINITION

PLEASE ENTER THE RECEIVE QUEUE FILE NAME! RCVQ
```

Figure 21

The simulator requests the name of the transmit queue file: "XMTQ".

```
2780/3780 SIMULATOR
LINE...      CLOSED      XMT QUEUE FILE.. NONE
MODE...      INACTIVE    XMT BATCH...     000
ACTIVE DATA FILE.. NONE    PRINT DEVICE...  PRINTER 1
ACTIVE DEVICE..  NONE    ALT PRINT DEVICE.. DISK
LAST REC NUMBER.. 0000    PUNCH DEVICE...  DISK
RCV QUEUE FILE..  RCVQ    ALT PUNCH DEVICE.. DISK
                **STATUS MESSAGES**
                SIGNONREC...

1)
2)
3)
4)
5)

*QUEUE FILE DEFINITION

PLEASE ENTER THE TRANSMIT QUEUE FILE NAME! XMTQ
```

```

2780/3780 SIMULATOR

LINE...      CLOSED      XMT QUEUE FILE... XMTQ
MODE...      INACTIVE    XMT BATCH...      000
ACTIVE DATA FILE.. NONE      PRINT DEVICE...   PRINTER 1
ACTIVE DEVICE... NONE      ALT PRINT DEVICE.. DISK
LAST REC NUMBER... 0000    PUNCH DEVICE...  DISK
RCV QUEUE FILE...  RCVQ     ALT PUNCH DEVICE.. DISK
                **STATUS MESSAGES**
                SIGNONREC...

1)
2)
3)
4)
5)

                                MODE MENU
1) DEFINE TERMINAL CHARACTERISTICS
2) FILE QUEUE DEFINITION
3) SIGNON DEFINITION
4) START COMMUNICATIONS
5) SET DATE/TIME

ENTER MODE NUMBER DESIRED ! 3

```

Figure 22

The user selects the signon definition mode in which proper signon record is selected.

```

2780/3780 SIMULATOR

LINE...      CLOSED      XMT QUEUE FILE... XMTQ
MODE...      INACTIVE    XMT BATCH...      000
ACTIVE DATA FILE.. NONE      PRINT DEVICE...   PRINTER 1
ACTIVE DEVICE... NONE      ALT PRINT DEVICE.. DISK
LAST REC NUMBER... 0000    PUNCH DEVICE...  DISK
RCV QUEUE FILE...  RCVQ     ALT PUNCH DEVICE.. DISK
                **STATUS MESSAGES**
                SIGNONREC...

1)
2)
3)
4)
5)

*SIGNON MENU
1) * .. SIGNON 1
555-1212 COMMENTS GO HERE

PLEASE ENTER THE SIGNON NUMBER DESIRED! 1

```

Figure 23

Signon menus are displayed. In this case, the user specifies that no more are to be displayed. (Maximum of three per screen.) The signon number desired signifies the first record transmitted to the remote mainframe.

```

2780/3780 SIMULATOR

LINE...      CLOSED      XMT QUEUE FILE... XMTQ
MODE...      INACTIVE    XMT BATCH...      000
ACTIVE DATA FILE.. NONE      PRINT DEVICE...   PRINTER 1
ACTIVE DEVICE... NONE      ALT PRINT DEVICE.. DISK
LAST REC NUMBER... 0000    PUNCH DEVICE...  DISK
RCV QUEUE FILE...  RCVQ     ALT PUNCH DEVICE.. DISK
                **STATUS MESSAGES**
                SIGNONREC... * .. SIGNON 1

1)
2)
3)
4)
5)

                                MODE MENU
1) DEFINE TERMINAL CHARACTERISTICS
2) FILE QUEUE DEFINITION
3) SIGNON DEFINITION
4) START COMMUNICATIONS
5) SET DATE/TIME

ENTER MODE NUMBER DESIRED ! 4

```

Figure 24

The user begins simulator execution by specifying MODE 4 "START COMMUNICATIONS."